# CRLibre | Instrucciones de uso e instalación de API de Facturación electrónica

Contenido

[CRLibre | Instrucciones de uso e instalación de API de Facturación electrónica 1](#_Toc517258482)

[Objetivo 2](#_Toc517258483)

[Instalación del Api 2](#_Toc517258484)

[Requerimientos mínimos y paquetes necesarios 2](#_Toc517258485)

[Introducción al api 2](#_Toc517258486)

[¿Por qué está separado en 2 secciones? 3](#_Toc517258487)

[Repositorio publico GitHUB 3](#_Toc517258488)

[Módulos del API para Facturación Electronica 3](#_Toc517258489)

[Instalación de Base de Datos 4](#_Toc517258490)

[Instalación de los paquetes necesarios 4](#_Toc517258491)

[Configuración del settings.php 5](#_Toc517258492)

[Primera petición al API 5](#_Toc517258493)

[Creación de Key de encriptación 6](#_Toc517258494)

[Uso de Módulos del API 6](#_Toc517258495)

[Uso de los modulos users\_openAccess 8](#_Toc517258496)

[Uso de los modulos users\_loggedIn 8](#_Toc517258497)

[Creación de Usuario 8](#_Toc517258498)

[Loging en el API 9](#_Toc517258499)

[LogOut del API 10](#_Toc517258500)

[Upload del certificado o llave criptográfica 11](#_Toc517258501)

[Solicitud de Token 12](#_Toc517258502)

[Solicitud de refrescar token 13](#_Toc517258503)

[Creación de Clave para los XML de Factura Electrónica 15](#_Toc517258504)

[Creación de Clave para Nota de Crédito 16](#_Toc517258505)

[Creación de Clave para Nota de Debito 18](#_Toc517258506)

[Creación de clave para Mensaje Aceptación (Aceptación total, Parcialmente y Rechazo) 19](#_Toc517258507)

[Creación de XML Factura Electrónica 22](#_Toc517258508)

[Creación de XML Nota de Crédito 29](#_Toc517258509)

[Creación de XML Nota de Debito 36](#_Toc517258510)

[Firmado del XML Factura Electrónica 43](#_Toc517258511)

[Firmado del XML Nota de Crédito 44](#_Toc517258512)

[Firmado del XML Nota de Debito 45](#_Toc517258513)

[Firmado del XML Mensaje de Aceptación (Pendiente solucionar error) 46](#_Toc517258514)

[Envió a Hacienda del XML de Factura Electrónica, Notas de Credito, Notas de Debito 47](#_Toc517258515)

[Envió a Hacienda del XML de Mensaje Aceptación (Aceptación total, Parcialmente y Rechazo) 49](#_Toc517258516)

[Comprobación de documentos 49](#_Toc517258517)

[Info extra y modulos a corregir o pendientes de cambios: 50](#_Toc517258518)

[Pendientes: 50](#_Toc517258519)

## Objetivo

El objetivo de este proyecto es poder brindar a la comunidad una herramienta capas de sobrellevar la comunicación con la plataforma de Hacienda de una manera más sencilla de entender.

## Instalación del Api

### Requerimientos mínimos y paquetes necesarios

* Php > 5.5.0
* MySQL o Mariadb
* Instalación librería curl
* Instalación php-XML

Introducción al api

Primeramente, hay que comprender la estructura o funcionamiento del API, ya que está funciona como una sólida base que nos permite extender y crear sobre ella por medio de módulos.

En este caso, el API cuenta con los módulos para realizar las diferentes tareas necesarias para la conexión con el Ministerio de Hacienda, así mismo como lo necesario para formatear la información y realizar las solicitudes necesarias. Lo único que es necesario, es tener la información de cada documento para realizar los diferentes procedimientos.

Es necesario identificar las 2 carpetas que estaremos utilizando. La primera es ‘www’, esta carpeta debe estar en una sección púbica de su servidor y es nuestra puerta de entrada a los diferentes módulos.

Dentro de esta carpeta se encuentra un documento en Php que se llama api.php. A este documento es al que se le deben hacer las diferentes solicitudes, ya sean POST, GET o por terminal.

Adicional se encuentra otro documento ‘settings.php’, en este se debe ingresar la configuración de la conexión a la base de datos y la ubicación de en donde se encuentra la segunda carpeta contigua a ‘www’, la cual tiene como nombre ‘api’.

### ¿Por qué está separado en 2 secciones?

La idea de esto es que el ‘www’ esté en una sección pública, y el resto del API con nuestros módulos, se encuentre en una sección en donde no sea abierto o accesible de manera pública. Esto nos brinda seguridad adicional en caso de que haya intromisión dentro del servidor y los documentos del API se encuentren resguardados por la seguridad del servidor o bien, de su sistema operativo.

Nosotros recomendamos utilizar un sistema Linux o basado en Unix ya que en entornos de servidores web son más ligeros, pero de igual manera se ha probado en IIS y el funcionamiento es óptimo.

Una vez que agreguemos los valores correctos en el ‘settings.php’, y tengamos nuestras carpetas en los lugares que queremos, podemos proceder a probar el API.

Un ejemplo de cómo se podría ver bajo un sistema Linux, sería algo como.

Ubicación del ‘www’:

/var/www/html/api.php

Ubicación del ‘api’

/var/www/api

De esta manera la ubicación del api en el settings.php línea 38, se definiría como:

$config['modules']['coreInstall'] = "../api/";

### Repositorio publico GitHUB

El Api se encuentra en un repositorio público de GitHub, en este repositorio pueden encontrar el Api completa bajo la URL:

https://github.com/CRLibre/API\_Hacienda

### Módulos del API para Facturación Electrónica

* Creación de Usuarios
* Login en el API
* Upload del certificado o llave criptográfica
* Solicitud de token
* Solicitud de refrescar token
* Creación de Clave para los XML de Factura Electrónica
* Creación de Clave para Nota de Crédito
* Creación de Clave para Nota de Débito
* Creación de clave para Mensaje Aceptación (Aceptación total, Parcialmente y Rechazo)
* Creación de XML Factura Electrónica
* Creación de XML Nota de Crédito
* Creación de XML Nota de Debito
* Firmado del XML Factura Electrónica
* Firmado del XML Nota de Crédito
* Firmado del XML Nota de Debito
* Firmado del XML Mensaje de Aceptación (Pendiente solucionar error)
* Envío a Hacienda del XML de Factura Electrónica
* Envío a Hacienda del XML de Nota de Debito
* Envío a Hacienda del XML de Nota de Crédito
* Envío a Hacienda del XML de Mensaje Aceptación (Aceptación total, Parcialmente y Rechazo)
* Creación de Clave para los XML de Factura Electrónica
* Creación de Clave para los XML de Nota de Crédito
* Creación de Clave para los XML de Nota de Debito
* Creación de Clave para los XML de Mensaje Aceptación (Aceptación total, Parcialmente y Rechazo)
* Consulta de estado de los comprobantes

### Instalación de Base de Datos

Una vez instalado el motor de base de datos (MySql o Mariadb) se debe crear una base de datos, usted elige el nombre.

Cuando la base de datos esta creado debe de importar la base de datos limpia que se encuentra en la siguiente ruta:

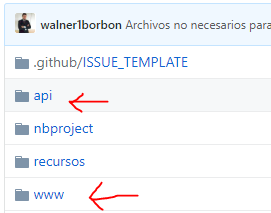
<https://github.com/CRLibre/API_Hacienda/tree/master/recursos>

### Instalación de los paquetes necesarios

https://github.com/CRLibre/API\_Hacienda

En la ruta del GitHub encontraras 2 carpetas necesarias:

1. Api
2. www



Esas son las 2 carpetas necesarias para operar el API

El contenido de la carpeta www es lo que va en el publico del servidor

Ejemplo:

Ubicación del ‘www’:

/var/www/html/api.php

/var/www/html/settings.php

/var/www/html/index.php

El index.php no es obligatorio, está ahí únicamente para redireccionar a una url en caso de que alguien quiera ingresar al server desde el navegador.

La carpeta Api es la que contiene los módulos necesarios para la operación.

Esta puede estar en cualquier lado del servidor siempre y cuando se defina correctamente la ubicación en el settings.php

Para este caso la colocar una carpeta antes del www/html

Ubicación del ‘api’

/var/www/api

### Configuración del settings.php

En el settings.php se definen las configuraciones iniciales del api.

* $config['modules']['coreInstall'] = "../api/";

Este es donde se define la ubicación del Api, aquí estamos definiendo que se encuentra una carpeta fuera, en caso de colocarla en otra ubicación debe también actualizar este parametro.

* $config['crypto']['key'] = "LkWfgWGQ/XhSd+ML13PEJsuecTHUPs9quAWGs1fMC9o=";

Esta es la KEY del proceso de encriptación, este método de encriptación es aes-256-cbc, este KEY se genera en el punto [Creación de Key de encriptación](#_Creación_de_Key)

* $config['db']['host'] = "localhost";

Este es el nombre del host donde esta ubicada la base de datos, en caso de que la base de datos este en otro servidor lo debe definir aquí

* $config['db']['user'] = "crlibre\_api";

Este es el usuario de la base de datos.

* $config['db']['pwd'] = 'S3syyZNw7woqFrm';

El password del usuario que tiene permisos sobre la base de datos.

* $config['db']['name'] = "crlibre\_api\_demo";

Este es el nombre de la base de datos

* $config['users']['sessionLifetime'] = 1000;

Esto es el tiempo que van a ser válidas las sesiones de los usuarios, esto parámetro se define en segundos por lo que debe ser tomado en cuenta a la hora de confiuracion

### Primera petición al API

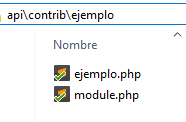
Las Peticiones al Api se realizan en Base al url o IP donde se encuentra el sitio, para mis efectos yo utilizare el API que se encuentra en demo de CRLibre.

La URL es <http://api-demo.crlibre.org/api.php>

Para el uso del Api tenemos que enviar un key=Value

Para nuestro primer test se va a llamar a una función de ejemplo en el modulo ejemplo

Este Modulo se llama ejemplo, dentro de module.php existe un array con rutas, esas rutas hacen referencia a funciones ubicadas ya sea en el mismo archivo php o en un file aparte de extencion .php, esto lo veremos a profundidad más adelante, ahorita nos enfocaremos en el método hola

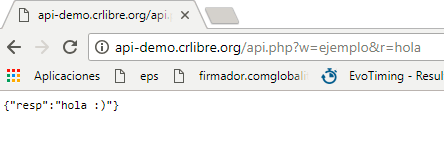


Para realizar este test vamos a enviarle al api 2 parámetros, un W que es el nombre del modulo (Mejor dicho el nombre de la carpeta) y un R que es el método que ejecutara dentro del module.php

Por un GET el api vamos a realizar la consulta, esta URL se puede abrir en cualquier navegador.

http://api-demo.crlibre.org/api.php?w=ejemplo&r=hola

Esto nos va a retornar un saludo desde el modulo hola :)



### Creación de Key de encriptación

El Key de encriptación es necesario para el settings.php, si no se genera un key y se define en el settings.php, el key sería el que trae el API y seria inseguro dejarlo así.

Para generar un KEY debe de llamar al módulo de la siguiente manera:

La ubicación del api:

<http://api-demo.crlibre.org/api.php>

Parámetros de petición: w=crypto r=makeKey

De modo GET sería de la siguiente manera:

<http://api-demo.crlibre.org/api.php?w=crypto&r=makeKey>

La respuesta del api será de la siguiente manera:

{"resp":"uWvFndG2QwW+RHCvG96U5iSHpDwFPsg5o4Ot3FRs+Gs="}

Ese sería el valor a definir en el settings.

**Importante** Este valor solo se puede definir una única vez, ya que como la encriptación es simétrica, si cambia el key no podrá desencriptar lo que tenías encriptado antes.

## Uso de Módulos del API

El uso del API es totalmente versátil para realizar peticiones de tipo POST o de tipo GET, para efectos prácticos y más trasparentes yo utilizaré PostMan para mostrar el funcionamiento de los diferentes Módulos.

Los módulos del API son Carpetas como tales, para los módulos de contribución se deben ubicar en la siguiente ruta \api\contrib

Los otros módulos son meramente del Core del API.

Usaré el módulo de clave para mostrarle el uso y la estructura del módulo.

* Debe existir una carpeta con el nombre del módulo ubicada en \api\contrib
* Dentro de clave debe existir un archivo llamado module.php
* La función \_bootMeUp debe ser completada con el nombre del módulo, como este módulo es ejemplo el \_bootMeUp se debe llamar ejemplo\_bootMeUp
* La función \_init() de igual manera debe llevar el nombre del módulo ejemplo\_init()
* El ejemplo\_init() es el que contiene un array de tareas.

En el siguiente código veremos que $paths es el Array con las funciones, cuando hacemos un GET enviamos un w este es el nombre del módulo o la carpeta, ahora vemos en $paths un r' => 'hola', esto quiere decir que si yo quiero llegar a usar ese módulo debo enviar un w=clave y un r=ejemplo el 'action' => ' module\_hola ', quiere decir que ejecutará la función module\_hola, 'access' => 'users\_openAccess' quiere decir que el usuario no requiere estar logueado para ejecutar la función, este puede ser users\_openAccess o users\_loggedIn en el caso de requerir logueo, 'file' => 'ejemplo.php' quiere decir que la función module\_hola se encuentra en el file ejemplo.php ubicado en la carpeta ejemplo y esa función será la que use los parámetros enviados y deberá retornar el resultado.



En el caso que queramos usar el método con parámetros les muestro el ejemplo de un\_usuario, este va a ejecutar la acción un\_usuario ubicada en ejemplo.php, pero este caso vamos a pedir unos parámetros que son requeridos "req" => true

Vamos a pedir un nombre y apellido

Por lo tanto, la consulta para llegar a ese modulo seria la siguiente:

http://api-demo.crlibre.org/api.php?w=ejemplo&r=un\_usuario&nombre=Walner&apellido=Borbon

### Uso de los modulos users\_openAccess

Para los módulos users\_openAccess no requiere enviar parámetros de sesión, únicamente los parámetros básicos del modulo.

### Uso de los modulos users\_loggedIn

Los módulos users\_loggedIn si se requiere enviar 2 parámetros especiales.

El primero es **iam** este es el userName del usuario

El segundo es **sessionKey** este sesión key es el que se genera en el módulo [Loging en el API](#_Loging_en_el) o cuando se crea un usuario por primera vez.

**En caso de que usted tenga definidos los módulos recuerde siempre enviar el iam y el sessionKey**

### Creación de Usuario

La creación de usuarios se encuentra en el modulo llamado users este esta en \api\modules\users

Para ver las funciones que tiene el modulo basta con abrir el archivo module.php

Para crear un usuario se debe enviar los siguientes parámetros:

w = users

r= users\_register

fullName = Esto es el nombre completo del Usuario

userName = Esto es el nombre de usuario

email = Esto es el correo del usuario

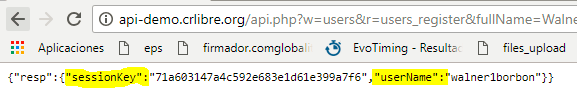
about = Esta es una informacion extra del usuario

country= Esta es informacion de Ubicacion

pwd= Esta es la contraseña para el usuario

El GET completo seria el siguiente:

<http://api-demo.crlibre.org/api.php?w=users&r=users_register&fullName=Walner%20Borbon&userName=walner1borbon&email=walner1borbon@gmail.com&about=otro%20Usuaruio&country=CR&pwd=123>



Esto nos retorna una sessionKey, esa sesiónKey es la que nos identifica como un usuario logueado en el API, posteriormente veremos cómo se loguea un usuario en el API para así obtener una nueva sessionKey.

### Loging en el API

Dado que el api utiliza 2 tipos de validación de usuarios ( users\_openAccess o users\_loggedIn ), en el caso de los modulos users\_openAccess no se requiere un sessionKey para ejecutarlos, pero en los que son users\_loggedIn si se requiere enviar al API una autentificación, por lo que en este paso le explicare la manera de obtener una.

Para ello se debe de interactuar con el modulo users pero el método que vamos a invocar será el users\_log\_me\_in

Para solicitar una nueva sessionKey se deben enviar los siguientes valores:

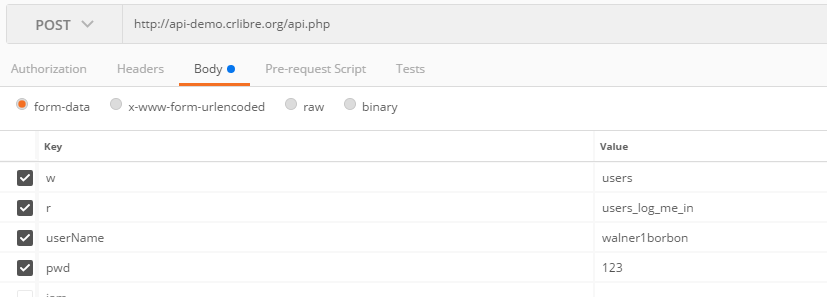
w = users

r = users\_log\_me\_in

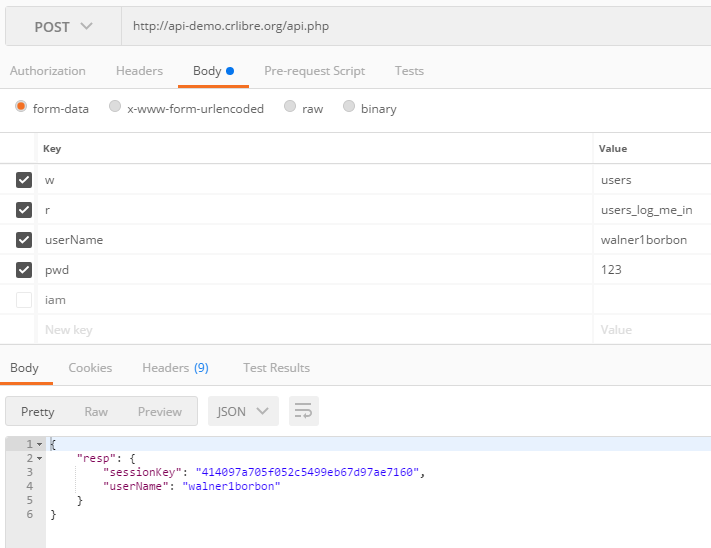
userName = userName previamente registrado

pwd= un password registrado

Parámetros enviados:



Al ejecutar esta petición el API nos devolverá lo siguiente:



Ese sessionKey es el que se debe de enviar a los métodos que son de tipo users\_loggedIn

### LogOut del API

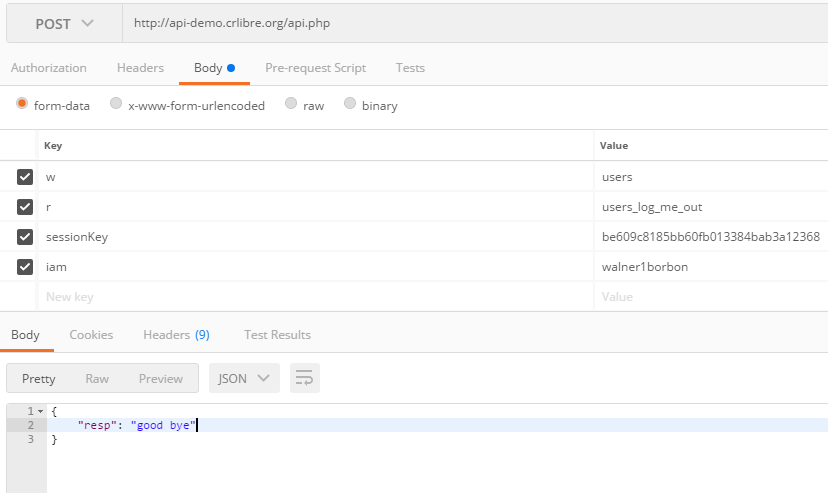
El api tiene la función de users\_log\_me\_out para cerrar la sesión, los parámetros a enviar son:

w:users

r:users\_log\_me\_out

sessionKey:be609c8185bb60fb013384bab3a12368

iam:walner1borbon



### Upload del certificado o llave criptográfica

El certificado será el que utilicemos para firmar los XML, este método solicita que el usuario este logueado, para lo que es necesario enviar un sessionKey, usaremos el que obtuvimos en el modulo anterios.

Para subir el certificado debe enviar los siguientes valores:

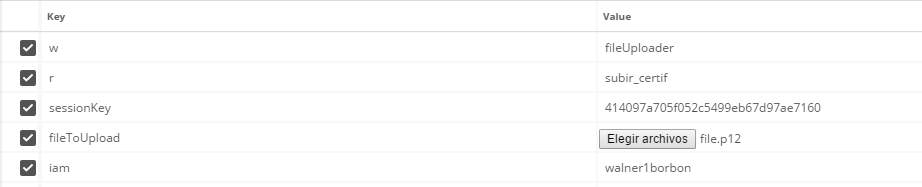
w = fileUploader

r = subir\_certif

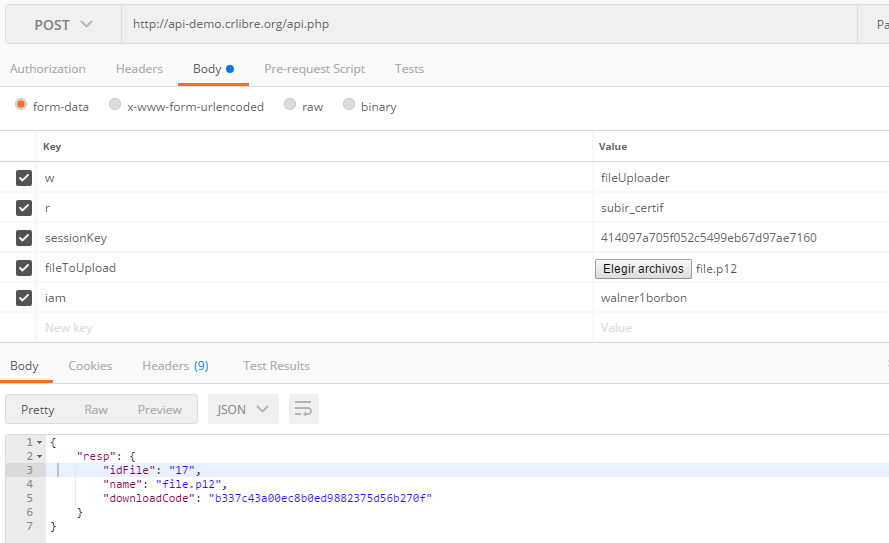
sessionKey= Esta la obtenemos previamente

fileToUpload= Este es el archivo p12 es necesario que sea en . p12 con la p en minúscula

iam su nombre de usuario



Esto nos va a retornar lo siguiente



Esa información es idFile es el id del file en db, name es el nombre del archivo y downloadCode es el código que se utilizara para firmar los documentos.

### Solicitud de Token

Para solicitar el Token se debe de enviar 6 datos:

w = token

r = gettoken

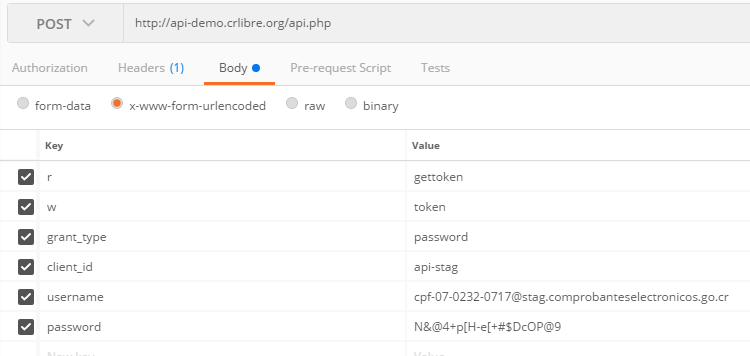
grant\_type = password para solicitar un token nuevo, mas adelante refresh\_token

client\_id = api-stag api-stag en sambox y api-prod para produccion

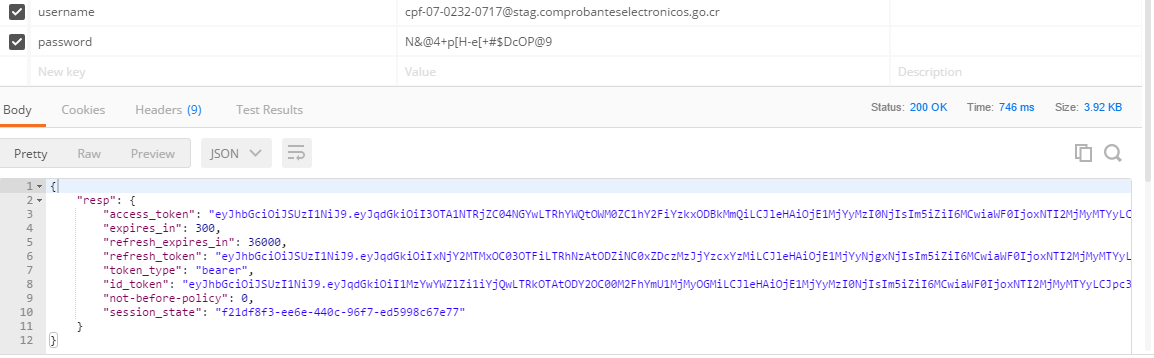
username= usuario de hacienda sambox o produccion

password Password de hacienda sambox o producción

La parte de username y password se estará modificando en las siguientes versiones.



Al enviar esto el Api responderá lo siguiente:



La sección del Access\_Token es lo importante, ya que eso es lo que enviaremos a hacienda con el resto de infomacion para el envio de los comprobantes y consultas.

El token expira en 300s y el refresh token funciona durante 36000s por lo que si vamos a refrescar el token debe ser antes de ese tiempo.

### Solicitud de refrescar token

Para refrescar el token debemos de enviar menos paramentros.

w = token

r = refresh

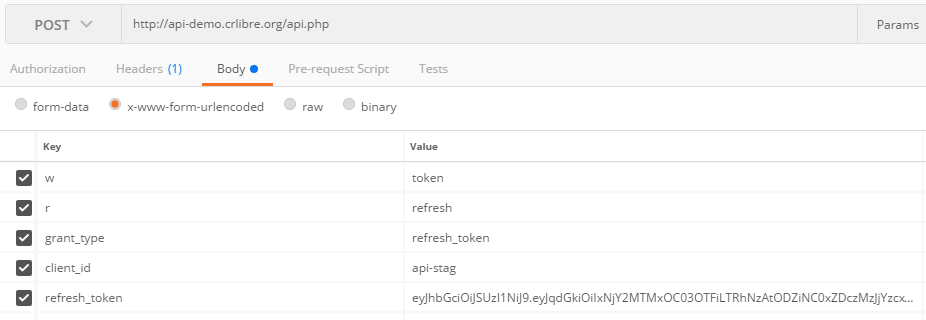
grant\_type= refresh\_token Esto dunciona solo 36000s después de solicitar el token

client\_id =api-stag api-stag en sambox y api-prod para produccion

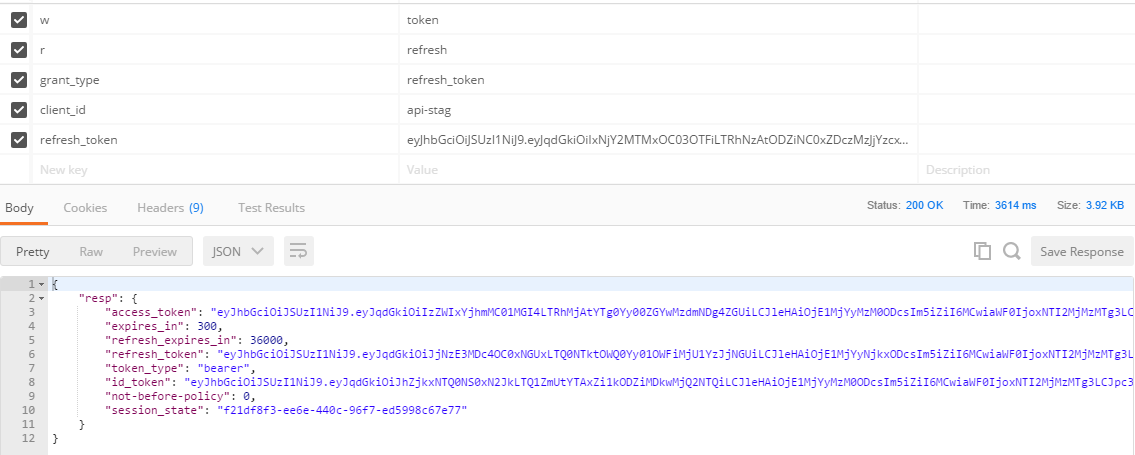
refresh\_token= el refresh\_token viene en la respuesta de solicitud del primer token



Por lo tanto la solicitud seria la siguiente



Y el resultado sería el siguiente, con un nuevo token y un nuevo refresh\_token:



### Creación de Clave para los XML de Factura Electrónica

Entre los parámetros a enviar se pueden enviar según el tipo de clave requerido, en este caso será para Factura electrónica (FE) pero voy a detallar los parámetros para las futuras construcciones de claves.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

tipoDocumento tiene las siguientes opciones:

FE Factura Electrónica

ND Nota de Débito

NC Nota de Crédito

TE Tiquete Electrónico

CCE Confirmación Comprobante Electrónico

CPCE Confirmación Parcial Comprobante Electrónico

RCE Rechazo Comprobante Electrónico

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

tipoCedula= físico o jurídico

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

situación=

normal En el caso de que la situación se normal

contingencia En el caso de que se tenga que enviar una clave de contingencia

sinInternet En el caso de que se tenga que hacer sin internet

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Para enviar a creación de la clave se envían los siguientes parámetros (En mi caso Cedula Física)

w=clave

r= clave

tipoCedula= físico o jurídico físico  
 cedula= Numero de Cedula 702320717

codigoPais= código del país 506

consecutivo= código de 10 números 1522773402

situacion= normal contingencia sinInternet normal

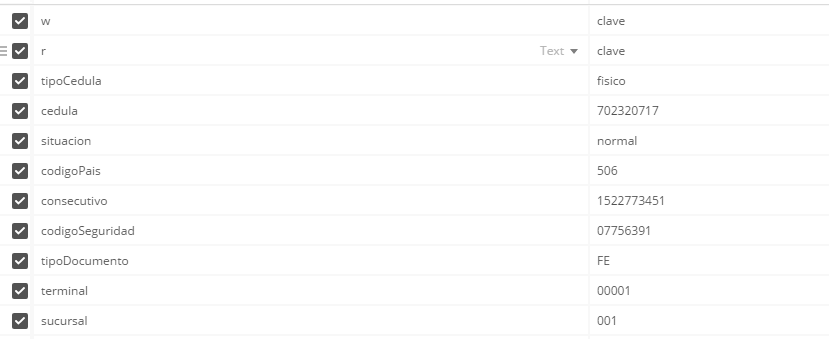
codigoSeguridad= código de 8 números 07756342

tipoDocumento= FE ND NC TE CCE CPCE RCE FE

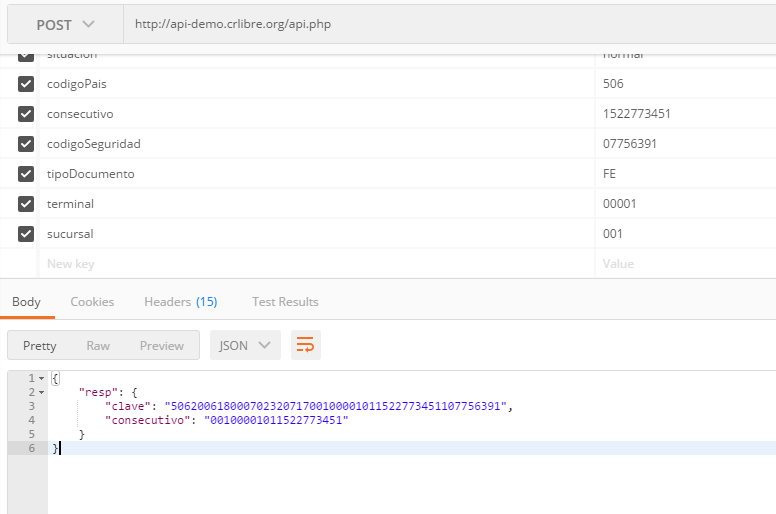
terminal: número de terminal 00001

sucursal= número de sucursal 001

Nota: Si no envía la sucursal o la terminal el sistema asigna por default terminal=00001 y sucursal =001



El API devolverá 2 valores, uno es la clave solicitada y el otro es un consecutivo, el consecutivo y la clave debe de ser insertados en el XML, por esa razón el API le devuelve los 2 de una vez.



### Creación de Clave para Nota de Crédito

Para enviar a creación de la clave de Nota de Crédito se envían los siguientes parámetros (En mi caso Cedula Física)

w=clave

r=clave

tipoCedula= físico o jurídico físico  
 cedula= Numero de Cedula 702320717

codigoPais= código del país 506

consecutivo= código de 10 numeros 1522773402

situacion= normal contingencia sinInternet normal

codigoSeguridad= código de 8 números 07756342

tipoDocumento= FE ND NC TE CCE CPCE RCE NC

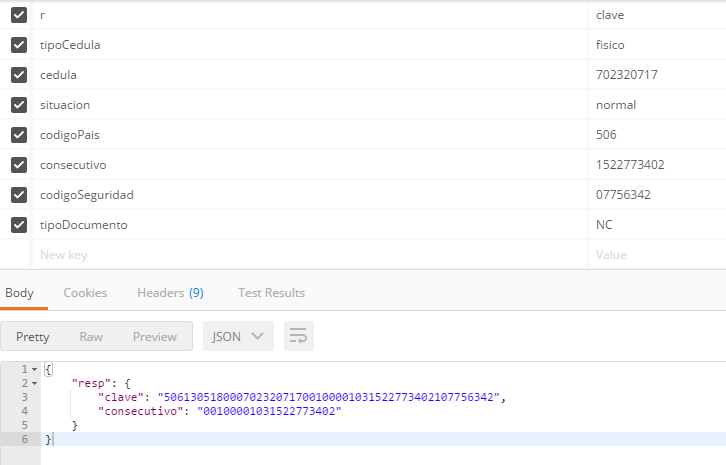
terminal: Numero de terminal 00001

sucursal= número de sucursal 001

Nota: Si no envía la sucursal o la terminal el sistema asigna por default terminal=00001 y sucursal =001



El API devolverá 2 valores, uno es la clave solicitada y el otro es un consecutivo, el consecutivo y la clave debe de ser insertados en el XML, por esa razón el API le devuelve los 2 de una vez.



### Creación de Clave para Nota de Debito

Para enviar a creación de la clave de Nota de Debito se envían los siguientes paramentros (En mi caso Cedula Fisica)

w=clave

r=clave

tipoCedula= físico o jurídico físico  
 cedula= Numero de Cedula 702320717

codigoPais= código del país 506

consecutivo= código de 10 numeros 1522773402

situacion= normal contingencia sininternet normal

codigoSeguridad= código de 8 numeros 07756342

tipoDocumento= FE ND NC TE CCE CPCE RCE ND

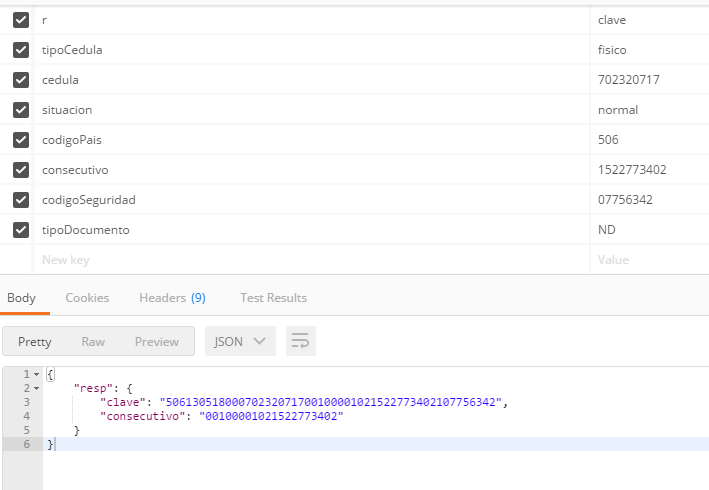
terminal: Numero de terminal 00001

sucursal= numero de sucursal 001

Nota: Si no envía la sucursal o la terminal el sistema asigna por default terminal=00001 y sucursal =001



El API devolverá 2 valores, uno es la clave solicitada y el otro es un consecutivo, el consecutivo y la clave debe de ser insertados en el XML, por esa razón el API le devuelve los 2 de una vez.



### Creación de clave para Mensaje Aceptación (Aceptación total, Parcialmente y Rechazo)

Para no hacer mas larga la documentación, explicare aquí el tema de Aceptacion total, parcial o de rechazo.

La creación del XML del mensaje de Aceptacion no implica que se cree una clave para el, lo que solicita es un numero consecutivo, ya que el XML que se genera lleva una clave de 50 caracteres, pero es la clave del documento que se esta aceptando.

Entonces para crear el XML de mensaje de aceptacion(total, parcial o rechazo) lo único que de requiere es la sección de consecutivo de la respuesta del API.

Para solicitarlo se envían los siguientes datos:

Para enviar a creación de la clave de Nota de Debito se envían los siguientes paramentros (En mi caso Cedula Fisica)

w=clave

r=clave

tipoCedula= físico o jurídico físico  
 cedula= Numero de Cedula 702320717

codigoPais= código del país 506

consecutivo= código de 10 numeros 1522773402

situacion= normal contingencia sininternet normal

codigoSeguridad= código de 8 numeros 07756342

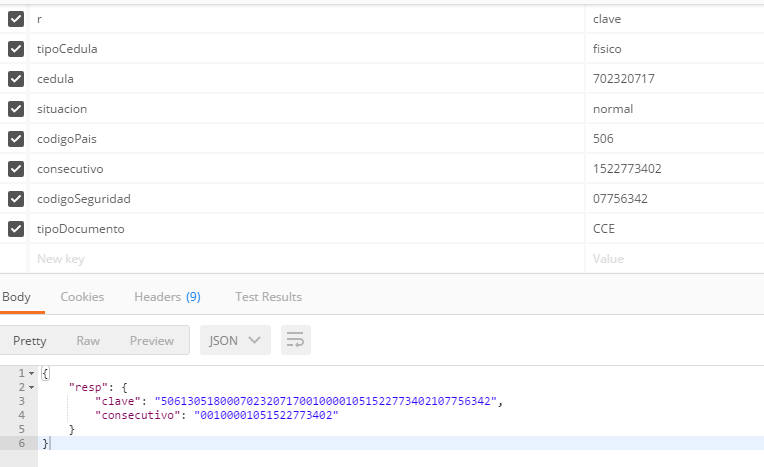
tipoDocumento= FE ND NC TE CCE CPCE RCE Este puede ser CCE CPCE RCE

terminal: Numero de terminal 00001

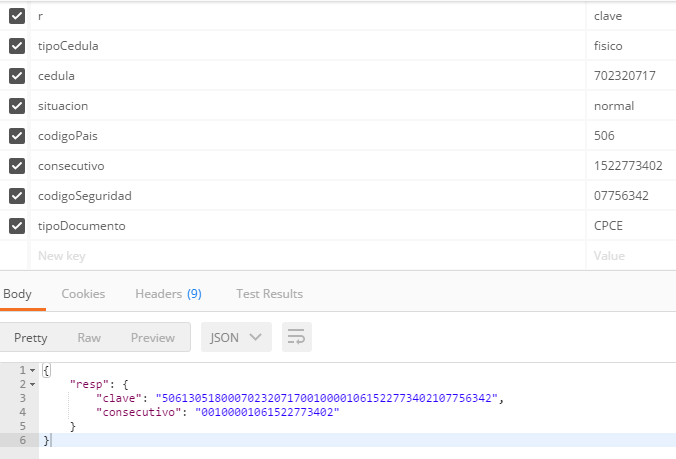
sucursal= numero de sucursal 001

Nota: Si no envía la sucursal o la terminal el sistema asigna por default terminal=00001 y sucursal =001

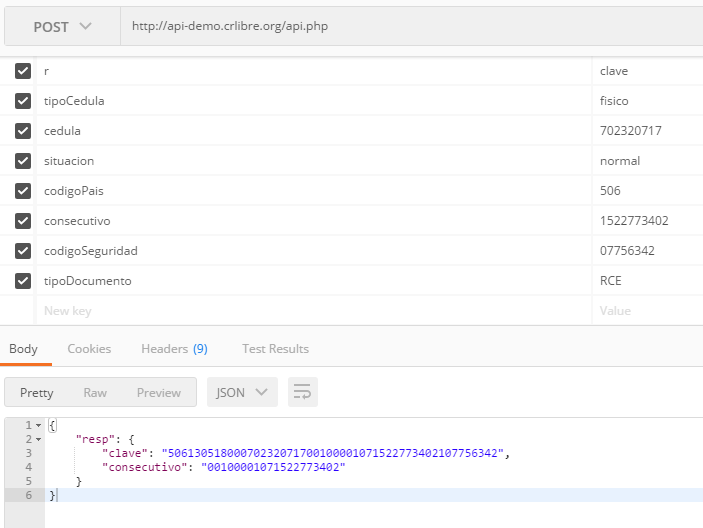
CCE | Confirmacion Comprabante Electrónico



CPCE | Confirmación Parcial Comprobante Electrónico



RCE | Rechazo Comprobante Electrónico



### Creación de XML Factura Electrónica

Para generar un XML de Factura electrónica se requieren varios datos, yo dejare aquí abajo los datos que voy a usar, todos son requeridos así, que no omitan ninguno.

Los Datos son:

w:genXML

r:gen\_xml\_fe

clave:50617061800070232071700100001011522773451107756391

consecutivo:00100001011522773451

fecha\_emision:2018-06-17T12:00:00-06:00

emisor\_nombre:Walner Borbon

emisor\_tipo\_indetif:01

emisor\_num\_identif:702320717

nombre\_comercial:Walner Borbon

emisor\_provincia:6

emisor\_canton:02

emisor\_distrito:03

emisor\_barrio:01

emisor\_otras\_senas: Frente a la escuela

emisor\_cod\_pais\_tel:506

emisor\_tel:64206205

emisor\_cod\_pais\_fax:506

emisor\_fax:00000000

emisor\_email:walner1borbon@gmail.com

receptor\_nombre:Walner Borbon

receptor\_tipo\_identif:01

receptor\_num\_identif:702320717

receptor\_provincia:6

receptor\_canton:02

receptor\_distrito:03

receptor\_barrio:01

receptor\_cod\_pais\_tel:506

receptor\_tel:84922891

receptor\_cod\_pais\_fax:506

receptor\_fax:00000000

receptor\_email:walner.borbon@hotmail.com

condicion\_venta:01

plazo\_credito:0

medio\_pago:01

cod\_moneda: CRC

tipo\_cambio:564.48

total\_serv\_gravados:0

total\_serv\_exentos:10000

total\_merc\_gravada:10000

total\_merc\_exenta:0

total\_gravados:10000

total\_exentos:10000

total\_ventas:20000

total\_descuentos:100

total\_ventas\_neta:19900

total\_impuestos:1170

total\_comprobante:21070

otros: Muchas gracias

detalles:{"1": {"cantidad":"1","unidadMedida":"Sp","detalle":"Impresora","precioUnitario":"10000","montoTotal":"10000","subtotal":"9900","montoTotalLinea":"9900","montoDescuento":"100","naturalezaDescuento":"Pronto pago"},"2":{"cantidad":"1","unidadMedida":"Unid","detalle":"producto","precioUnitario":"10000","montoTotal":"10000","subtotal":"10000","montoTotalLinea":"11170","impuesto":{"1": {"código":"01","tarifa":"11.7","monto":"1170"}}}}

El campo detalle debe enviarse con la siguiente estructura:

Estos son datos de ejemplo, no están correctamente sumados o restados, solo es para ejemplificar la estructura.

{

"1": {

"cantidad": "1",

"unidadMedida": "sp",

"detalle": "Impresora",

"precioUnitario": "10000",

"montoTotal": "1000",

"subtotal": "9900",

"montoTotalLinea": "12177",

"montoDescuento": "100",

"naturalezaDescuento": "Pronto pago",

"impuesto": {

"1": {

"código": "01",

"tarifa": "13",

"monto": "1287"

},

"2": {

"código": "02",

"tarifa": "10",

"monto": "990"

}

}

},

"2": {

"cantidad": "1",

"unidadMedida": "sp",

"detalle": "producto",

"precioUnitario": "10000",

"montoTotal": "10000",

"subtotal": "10000",

"montoTotalLinea": "11300",

"impuesto": {

"1": {

"código": "01",

"tarifa": "13",

"monto": "1300",

"exoneracion": {

"tipoDocumento": "01",

"numeroDocumento": "100",

"nombreInstitucion": "Ministerio de Hacienda",

"fechaEmision": "2016-09-26T13:00:00+06:00",

"montoImpuesto": "130",

"porcentajeCompra": "10"

}

}

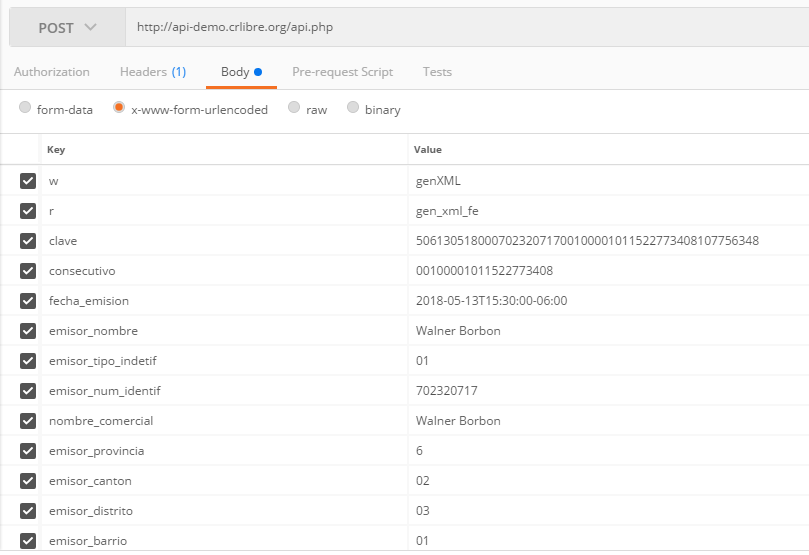
}

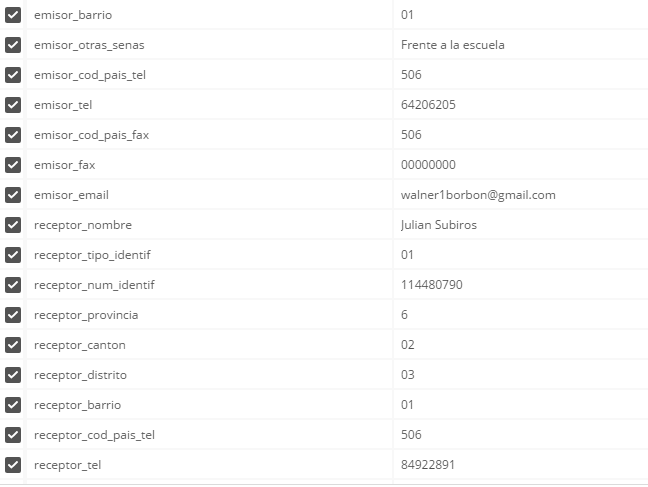
}

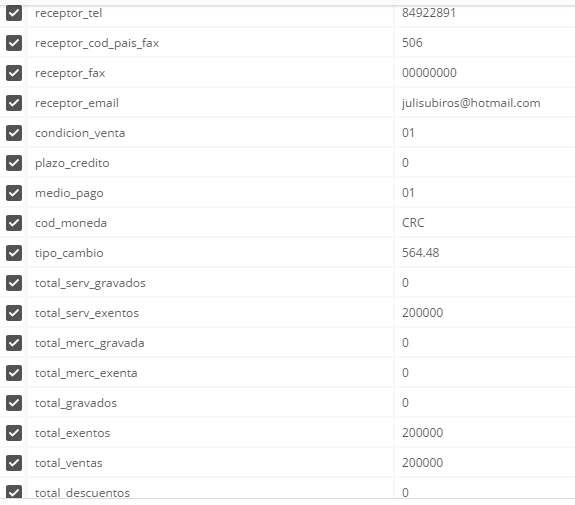
}

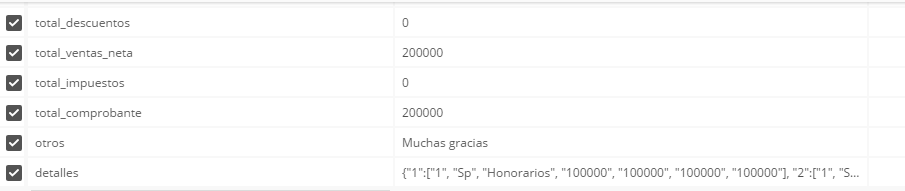
Esto es un ejemplo con la suma correcta de los campos

La línea 1 es con descuento y la segunda es con impuesto (Los descuentos pueden ser varios por lo que se debe definir un arreglo de impuestos por si son 2 impuestos)  
**//Paramentr detalles**:{    
//#Linea 1 **"1"**:{    
//Cantidad     **"cantidad"**:"1",  
//UnidadMedida **"unidadMedida"**:"Sp",  
//Detalle linea   **"detalle"**:"Impresora",  
//precio unidad   **"precioUnitario"**:"10000",  
//Monto total línea   **"montoTotal"**:"10000",  
//subtotal linea     **"subtotal"**:"9900",  
//Total de linea    **"montoTotalLinea"**:"9900",  
//Cantidad descuento   **"montoDescuento"**:"100",  
//Razon del descuento       **"naturalezaDescuento"**:"Pronto pago"  
//#Fin linea 1  },  
//#Linea 2  **"2"**:{    
//Cantidad       **"cantidad"**:"1",  
//Unidad Medida       **"unidadMedida"**:"Unid",  
//Detalle linea       **"detalle"**:"producto",  
//Precio unidad       **"precioUnitario"**:"10000",  
//Monto total línea      **"montoTotal"**:"10000",  
//Total de linea     **"subtotal"**:"10000",  
//Cantidad descuento      **"montoTotalLinea"**:"11170",  
//#IniciaImpuesto       **"impuesto"**:{    
//#Primer Impuesto          **"1"**:{    
//Codigo Impuesto             **"código"**:"01",  
//Tarifa Impuesto             **"tarifa"**:"11.7",  
//Monto Impuesto             **"monto"**:"1170"  
//#Cierra impuesto 1      }  
//#Cierra impuestos       }  
//#Cierra línea 2    }  
//##Cierra detalle }



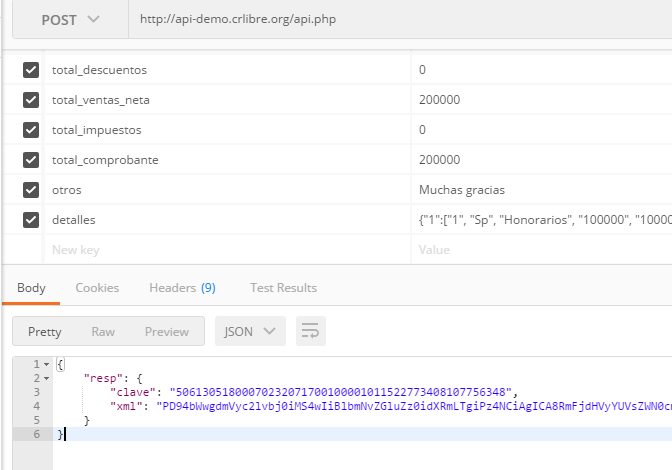






La respuesta del API será clave y XML, pero ese XML esta sin firmar y en base64

Por lo que deberá luego ser firmado.



### Creación de XML Nota de Crédito

Para generar un XML de Nota de credito se requieren varios datos, yo dejare aquí abajo los datos que voy a usar, todos son requeridos así, que no omitan ninguno.

Los Datos son:

w:genXML

r: gen\_xml\_nc

clave:50617061800070232071700100001011522773451107756391

consecutivo:00100001011522773451

fecha\_emision:2018-06-17T12:00:00-06:00

emisor\_nombre:Walner Borbon

emisor\_tipo\_indetif:01

emisor\_num\_identif:702320717

nombre\_comercial:Walner Borbon

emisor\_provincia:6

emisor\_canton:02

emisor\_distrito:03

emisor\_barrio:01

emisor\_otras\_senas: Frente a la escuela

emisor\_cod\_pais\_tel:506

emisor\_tel:64206205

emisor\_cod\_pais\_fax:506

emisor\_fax:00000000

emisor\_email:walner1borbon@gmail.com

receptor\_nombre:Walner Borbon

receptor\_tipo\_identif:01

receptor\_num\_identif:702320717

receptor\_provincia:6

receptor\_canton:02

receptor\_distrito:03

receptor\_barrio:01

receptor\_cod\_pais\_tel:506

receptor\_tel:84922891

receptor\_cod\_pais\_fax:506

receptor\_fax:00000000

receptor\_email:walner.borbon@hotmail.com

condicion\_venta:01

plazo\_credito:0

medio\_pago:01

cod\_moneda:CRC

tipo\_cambio:564.48

total\_serv\_gravados:0

total\_serv\_exentos:10000

total\_merc\_gravada:10000

total\_merc\_exenta:0

total\_gravados:10000

total\_exentos:10000

total\_ventas:20000

total\_descuentos:100

total\_ventas\_neta:19900

total\_impuestos:1170

total\_comprobante:21070

otros:Muchas gracias

detalles:{"1": {"cantidad":"1","unidadMedida":"Sp","detalle":"Impresora","precioUnitario":"10000","montoTotal":"10000","subtotal":"9900","montoTotalLinea":"9900","montoDescuento":"100","naturalezaDescuento":"Pronto pago"},"2":{"cantidad":"1","unidadMedida":"Unid","detalle":"producto","precioUnitario":"10000","montoTotal":"10000","subtotal":"10000","montoTotalLinea":"11170","impuesto":{"1": {"código":"01","tarifa":"11.7","monto":"1170"}}}}

infoRefeTipoDoc:01

infoRefeNumero:50615051800070232071700100001011522773426107756366

infoRefeFechaEmision:2018-05-15T16:00:00-06:00

infoRefeCodigo:03

infoRefeRazon:Falta de informacion

El campo detalle debe enviarse con la siguiente estructura:

Estos son datos de ejemplo, no están correctamente sumados o restados, solo es para ejemplificar la estructura.

{

"1": {

"cantidad": "1",

"unidadMedida": "sp",

"detalle": "Impresora",

"precioUnitario": "10000",

"montoTotal": "1000",

"subtotal": "9900",

"montoTotalLinea": "12177",

"montoDescuento": "100",

"naturalezaDescuento": "Pronto pago",

"impuesto": {

"1": {

"código": "01",

"tarifa": "13",

"monto": "1287"

},

"2": {

"código": "02",

"tarifa": "10",

"monto": "990"

}

}

},

"2": {

"cantidad": "1",

"unidadMedida": "sp",

"detalle": "producto",

"precioUnitario": "10000",

"montoTotal": "10000",

"subtotal": "10000",

"montoTotalLinea": "11300",

"impuesto": {

"1": {

"código": "01",

"tarifa": "13",

"monto": "1300",

"exoneracion": {

"tipoDocumento": "01",

"numeroDocumento": "100",

"nombreInstitucion": "Ministerio de Hacienda",

"fechaEmision": "2016-09-26T13:00:00+06:00",

"montoImpuesto": "130",

"porcentajeCompra": "10"

}

}

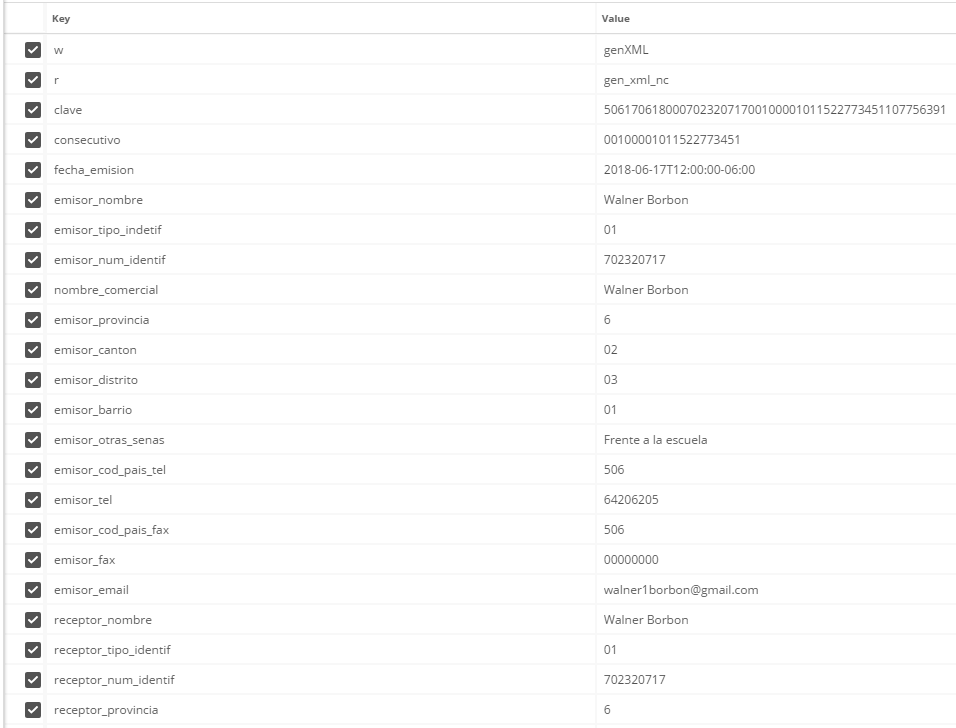
}

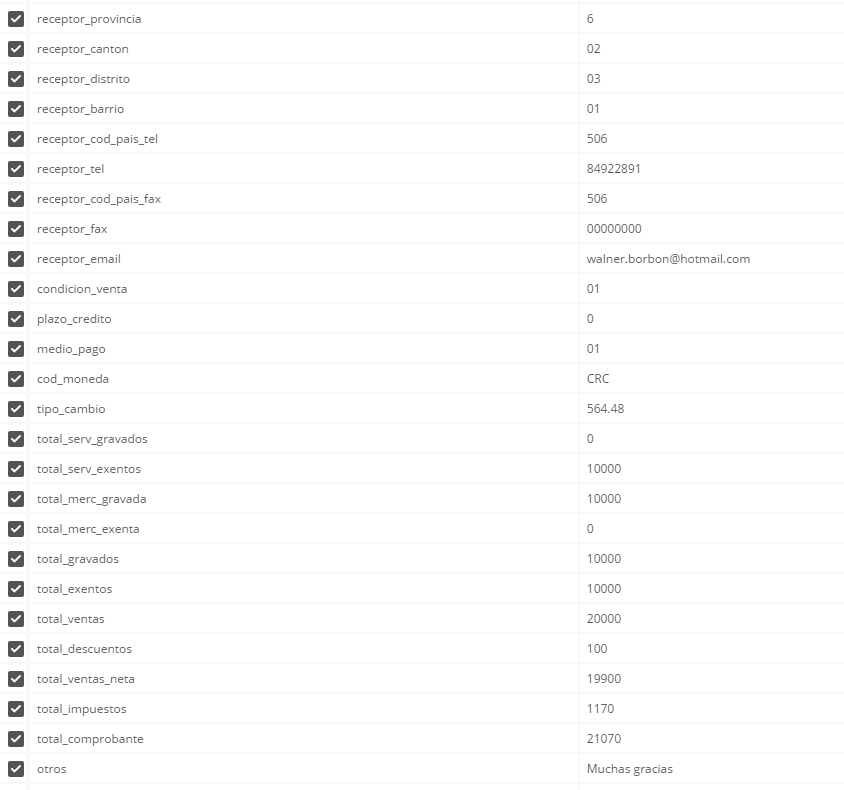
}

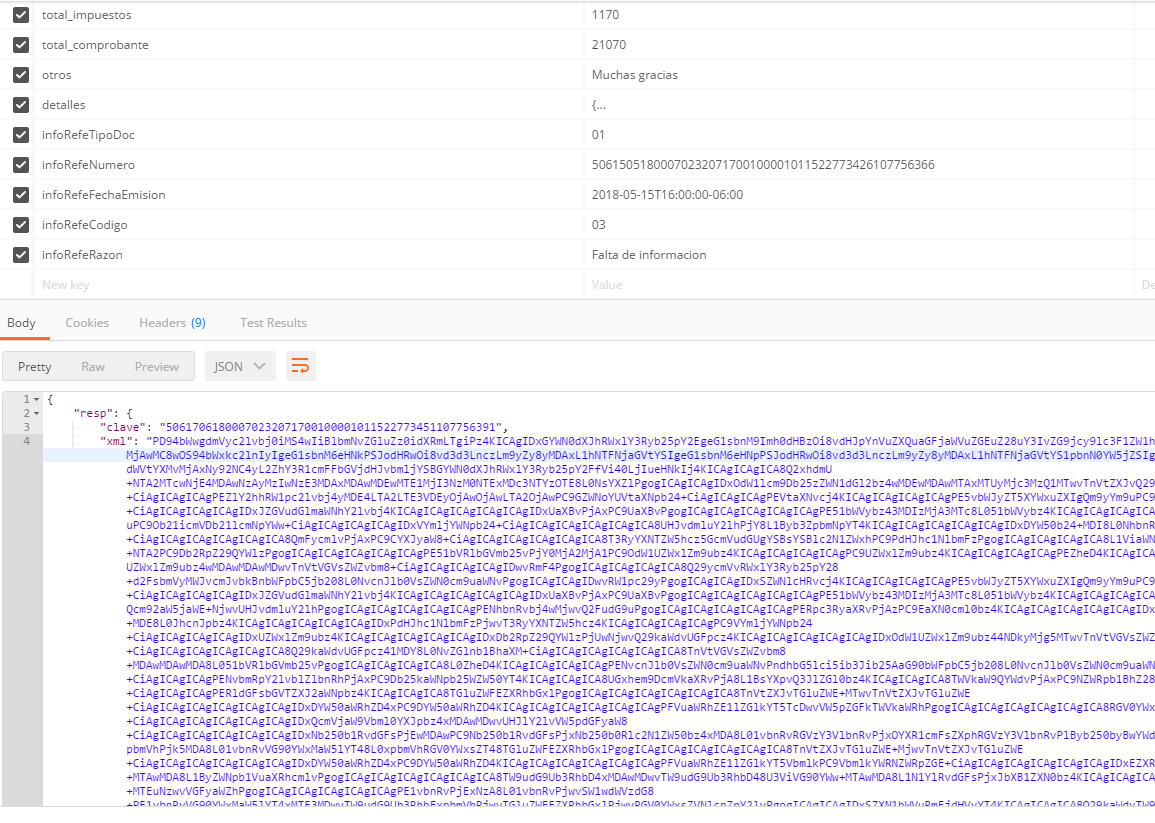
}

Esto es un ejemplo con la suma correcta de los compos

La línea 1 es con descuento y la segunta es con impuesto (Los descuentos pueden ser varios por lo que se debe definir un arreglo de impuestos por si son 2 impuestos)  
**//Paramentr detalles**:{    
//#Linea 1 **"1"**:{    
//Cantidad     **"cantidad"**:"1",  
//UnidadMedida **"unidadMedida"**:"Sp",  
//Detalle linea   **"detalle"**:"Impresora",  
//precio unidad   **"precioUnitario"**:"10000",  
//Monto total línea   **"montoTotal"**:"10000",  
//subtotal linea     **"subtotal"**:"9900",  
//Total de linea    **"montoTotalLinea"**:"9900",  
//Cantidad descuento   **"montoDescuento"**:"100",  
//Razon del descuento       **"naturalezaDescuento"**:"Pronto pago"  
//#Fin linea 1  },  
//#Linea 2  **"2"**:{    
//Cantidad       **"cantidad"**:"1",  
//Unidad Medida       **"unidadMedida"**:"Unid",  
//Detalle linea       **"detalle"**:"producto",  
//Precio unidad       **"precioUnitario"**:"10000",  
//Monto total línea      **"montoTotal"**:"10000",  
//Total de linea     **"subtotal"**:"10000",  
//Cantidad descuento      **"montoTotalLinea"**:"11170",  
//#IniciaImpuesto       **"impuesto"**:{    
//#Primer Impuesto          **"1"**:{    
//Codigo Impuesto             **"código"**:"01",  
//Tarifa Impuesto             **"tarifa"**:"11.7",  
//Monto Impuesto             **"monto"**:"1170"  
//#Cierra impuesto 1      }  
//#Cierra impuestos       }  
//#Cierra línea 2    }  
//##Cierra detalle }







### Creación de XML Nota de Debito

Para generar un XML de Nota Debito se requieren varios datos, yo dejare aquí abajo los datos que voy a usar, todos son requeridos así, que no omitan ninguno.

Los Datos son:

w:genXML

r: gen\_xml\_nd

clave:50617061800070232071700100001011522773451107756391

consecutivo:00100001011522773451

fecha\_emision:2018-06-17T12:00:00-06:00

emisor\_nombre:Walner Borbon

emisor\_tipo\_indetif:01

emisor\_num\_identif:702320717

nombre\_comercial:Walner Borbon

emisor\_provincia:6

emisor\_canton:02

emisor\_distrito:03

emisor\_barrio:01

emisor\_otras\_senas:Frente a la escuela

emisor\_cod\_pais\_tel:506

emisor\_tel:64206205

emisor\_cod\_pais\_fax:506

emisor\_fax:00000000

emisor\_email:walner1borbon@gmail.com

receptor\_nombre:Walner Borbon

receptor\_tipo\_identif:01

receptor\_num\_identif:702320717

receptor\_provincia:6

receptor\_canton:02

receptor\_distrito:03

receptor\_barrio:01

receptor\_cod\_pais\_tel:506

receptor\_tel:84922891

receptor\_cod\_pais\_fax:506

receptor\_fax:00000000

receptor\_email:walner.borbon@hotmail.com

condicion\_venta:01

plazo\_credito:0

medio\_pago:01

cod\_moneda:CRC

tipo\_cambio:564.48

total\_serv\_gravados:0

total\_serv\_exentos:10000

total\_merc\_gravada:10000

total\_merc\_exenta:0

total\_gravados:10000

total\_exentos:10000

total\_ventas:20000

total\_descuentos:100

total\_ventas\_neta:19900

total\_impuestos:1170

total\_comprobante:21070

otros:Muchas gracias

detalles:{"1": {"cantidad":"1","unidadMedida":"Sp","detalle":"Impresora","precioUnitario":"10000","montoTotal":"10000","subtotal":"9900","montoTotalLinea":"9900","montoDescuento":"100","naturalezaDescuento":"Pronto pago"},"2":{"cantidad":"1","unidadMedida":"Unid","detalle":"producto","precioUnitario":"10000","montoTotal":"10000","subtotal":"10000","montoTotalLinea":"11170","impuesto":{"1": {"código":"01","tarifa":"11.7","monto":"1170"}}}}

infoRefeTipoDoc:01

infoRefeNumero:50615051800070232071700100001011522773426107756366

infoRefeFechaEmision:2018-05-15T16:00:00-06:00

infoRefeCodigo:03

infoRefeRazon:Falta de informacion

Estos campos hacen referencia a la información del documento que se le genera la NC

// Tipo de documento de referencia. 01 Factura electronica, 02 Nota de //debito electronica, 03 nota de credito electronica, 04 Tiquete //electronico

infoRefeTipoDoc:01

// Clave documento de referencia

infoRefeNumero:50615051800070232071700100001011522773426107756366

//Fecha de emisión del documento

infoRefeFechaEmision:2018-05-15T16:00:00-06:00

//01 Anula documento de referencia, 02 Corrige texto de documento de //referencia, 03 Corrige monto, 04 Referencia a otro documento, 99 Otros

infoRefeCodigo:03

//Informacion del porque esta generando la NC

infoRefeRazon:Falta de informacion

El campo detalle debe enviarse con la siguiente estructura:

Estos son datos de ejemplo, no están correctamente sumados o restados, solo es para ejemplificar la estructura.

{

"1": {

"cantidad": "1",

"unidadMedida": "sp",

"detalle": "Impresora",

"precioUnitario": "10000",

"montoTotal": "1000",

"subtotal": "9900",

"montoTotalLinea": "12177",

"montoDescuento": "100",

"naturalezaDescuento": "Pronto pago",

"impuesto": {

"1": {

"código": "01",

"tarifa": "13",

"monto": "1287"

},

"2": {

"código": "02",

"tarifa": "10",

"monto": "990"

}

}

},

"2": {

"cantidad": "1",

"unidadMedida": "sp",

"detalle": "producto",

"precioUnitario": "10000",

"montoTotal": "10000",

"subtotal": "10000",

"montoTotalLinea": "11300",

"impuesto": {

"1": {

"código": "01",

"tarifa": "13",

"monto": "1300",

"exoneracion": {

"tipoDocumento": "01",

"numeroDocumento": "100",

"nombreInstitucion": "Ministerio de Hacienda",

"fechaEmision": "2016-09-26T13:00:00+06:00",

"montoImpuesto": "130",

"porcentajeCompra": "10"

}

}

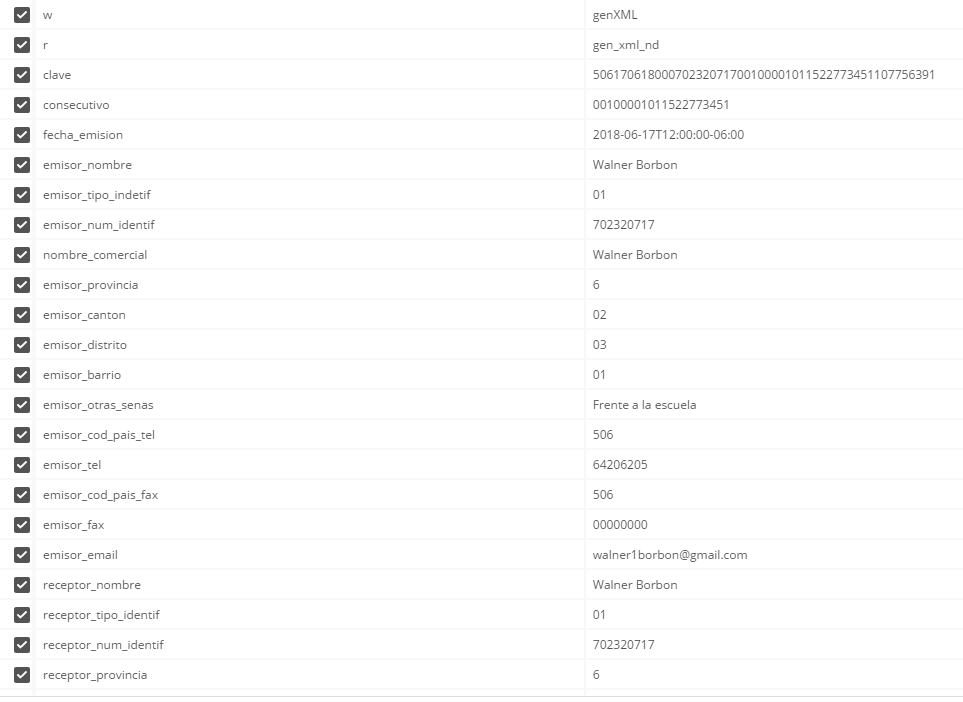
}

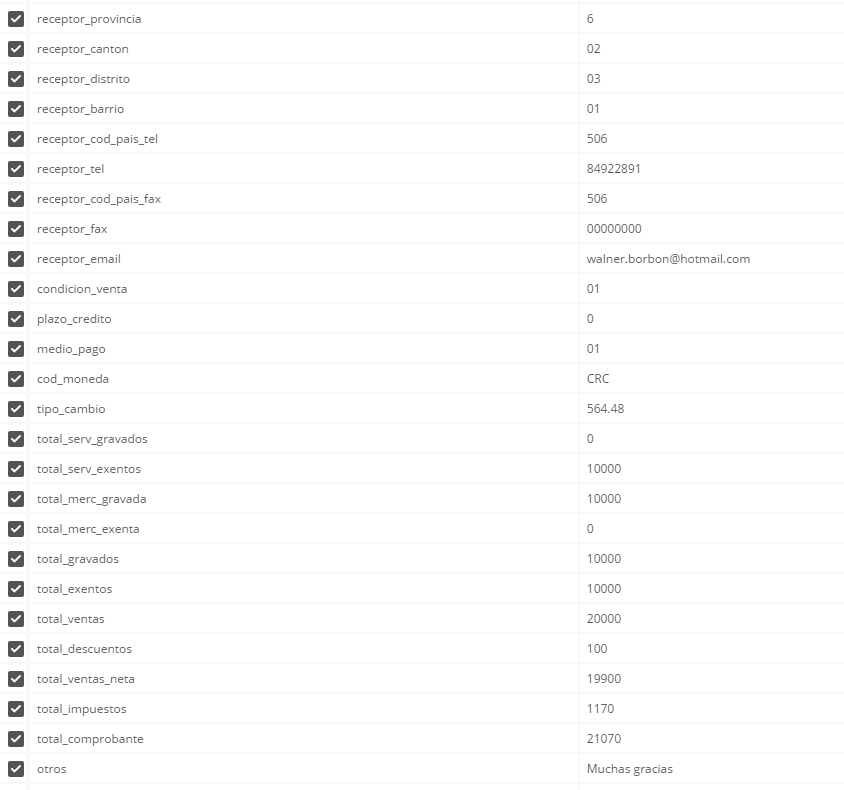
}

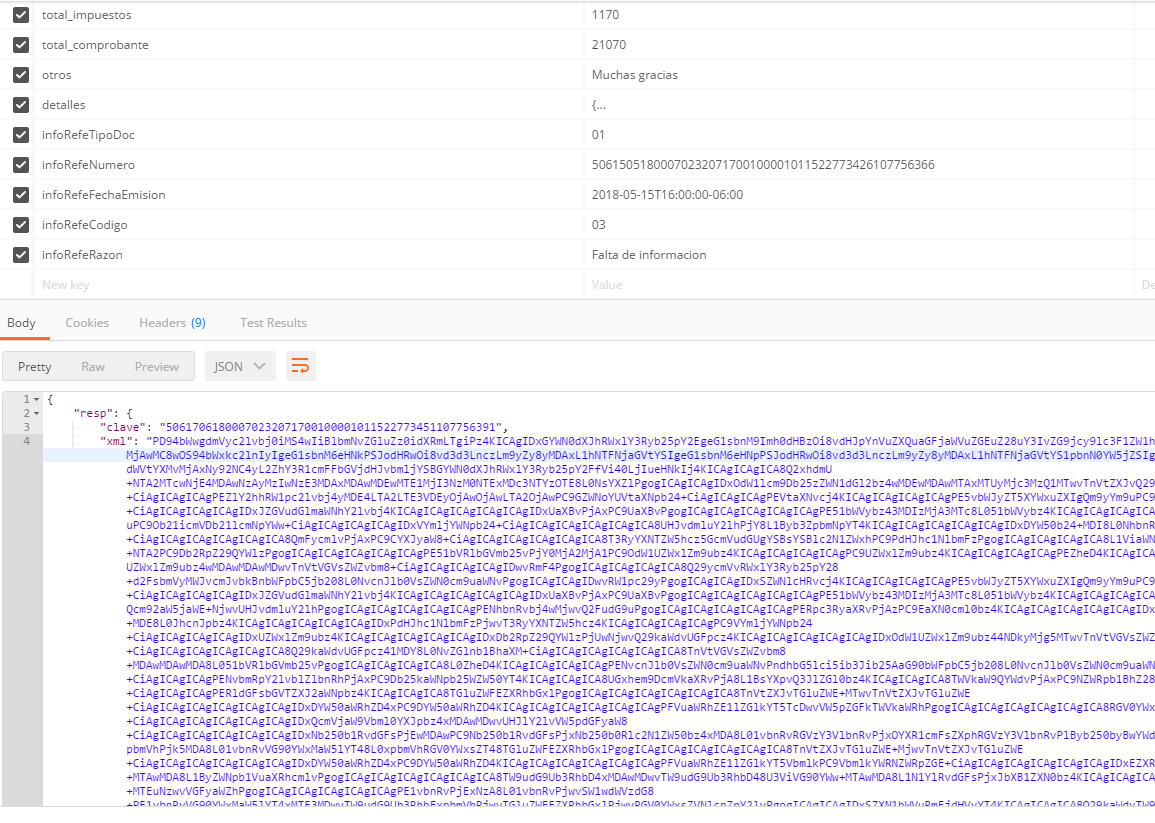
}

Esto es un ejemplo con la suma correcta de los compos

La línea 1 es con descuento y la segunta es con impuesto (Los descuentos pueden ser varios por lo que se debe definir un arreglo de impuestos por si son 2 impuestos)  
**//Paramentr detalles**:{    
//#Linea 1 **"1"**:{    
//Cantidad     **"cantidad"**:"1",  
//UnidadMedida **"unidadMedida"**:"Sp",  
//Detalle linea   **"detalle"**:"Impresora",  
//precio unidad   **"precioUnitario"**:"10000",  
//Monto total línea   **"montoTotal"**:"10000",  
//subtotal linea     **"subtotal"**:"9900",  
//Total de linea    **"montoTotalLinea"**:"9900",  
//Cantidad descuento   **"montoDescuento"**:"100",  
//Razon del descuento       **"naturalezaDescuento"**:"Pronto pago"  
//#Fin linea 1  },  
//#Linea 2  **"2"**:{    
//Cantidad       **"cantidad"**:"1",  
//Unidad Medida       **"unidadMedida"**:"Unid",  
//Detalle linea       **"detalle"**:"producto",  
//Precio unidad       **"precioUnitario"**:"10000",  
//Monto total línea      **"montoTotal"**:"10000",  
//Total de linea     **"subtotal"**:"10000",  
//Cantidad descuento      **"montoTotalLinea"**:"11170",  
//#IniciaImpuesto       **"impuesto"**:{    
//#Primer Impuesto          **"1"**:{    
//Codigo Impuesto             **"código"**:"01",  
//Tarifa Impuesto             **"tarifa"**:"11.7",  
//Monto Impuesto             **"monto"**:"1170"  
//#Cierra impuesto 1      }  
//#Cierra impuestos       }  
//#Cierra línea 2    }  
//##Cierra detalle }







### Firmado del XML Factura Electrónica

Para firmar un XML se debe de hacer uso del Token del certificado, el cual subimos en la sección de [Upload del certificado o llave criptográfica](#_Upload_del_certificado)

Los parámetros a enviar para poder firmar un XML son los siguientes:

w= signXML

r= signFE

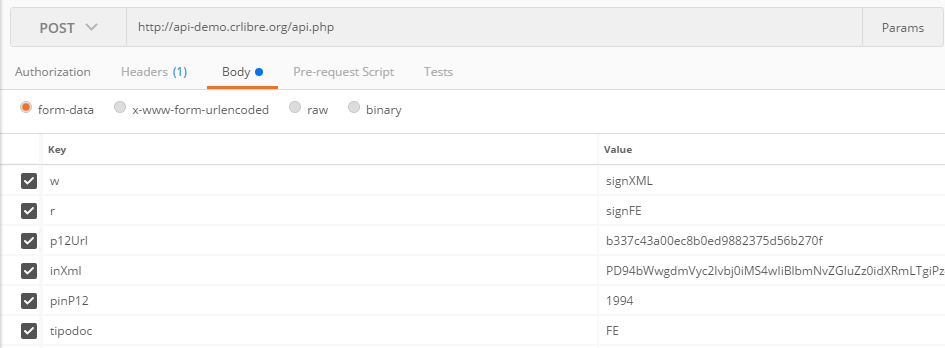
p12Url= Este es el código que se obtiene al subir el certificado

inXml= Este debe ser el XML que se va a firmar, pero debe ir en base64

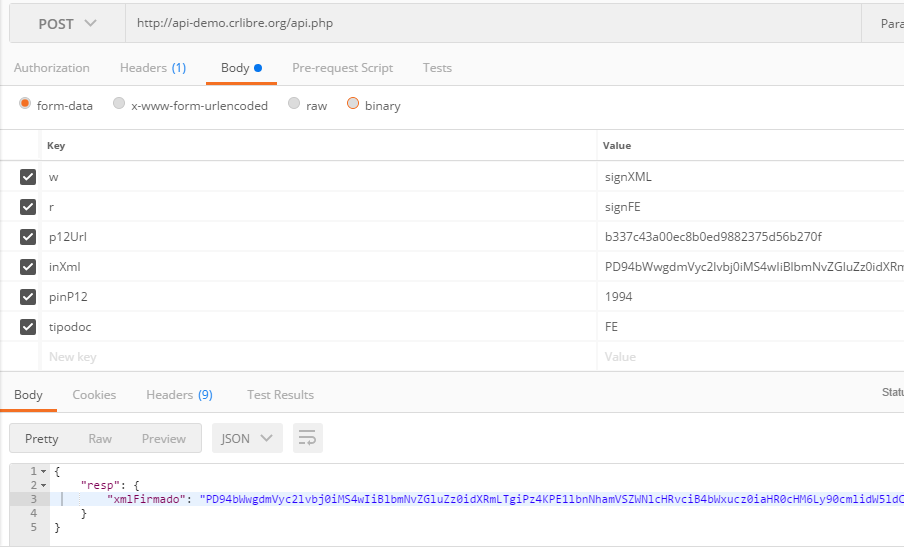
pinP12= Esta es la clave del certificado

tipodoc= Tipo de documento igual puede ser FE

b337c43a00ec8b0ed9882375d56b270f es el código que obtuvimos al subir el certificado.



El XML firmado viene en Base64, esto para que el parámetro se pase de una vez a envio a hacienda, la respuesta seria la siguiente:



### Firmado del xml Nota de Crédito

Para firmar un XML se debe de hacer uso del Token del certificado, el cual subimos en la sección de [Upload del certificado o llave criptográfica](#_Upload_del_certificado)

Los parámetros a enviar para poder firmar un XML son los siguientes:

w= signXML

r= signFE

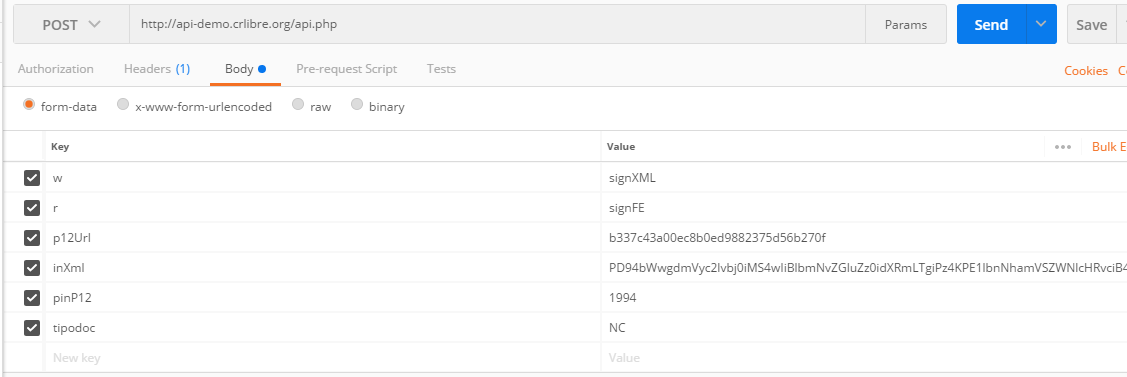
p12Url= Este es el código que se obtiene al subir el certificado

inXml= Este debe ser el XML que se va a firmar, pero debe ir en base64

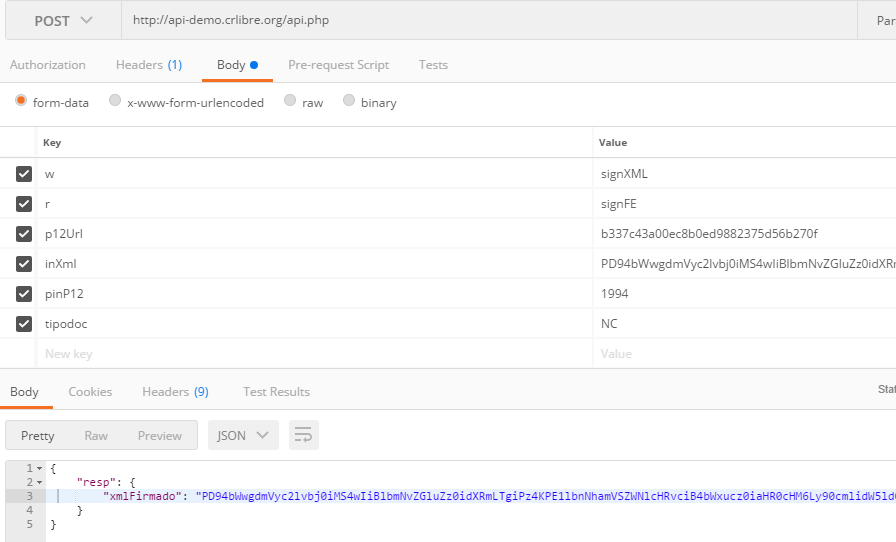
pinP12= Esta es la clave del certificado

tipodoc= Tipo de documento igual puede ser NC

b337c43a00ec8b0ed9882375d56b270f es el código que obtuvimos al subir el certificado.



El XML firmado viene en Base64, esto para que el parámetro se pase de una vez a envio a hacienda, la respuesta seria la siguiente:



### Firmado del XML Nota de Debito

Para firmar un XML se debe de hacer uso del Token del certificado, el cual subimos en la sección de [Upload del certificado o llave criptográfica](#_Upload_del_certificado)

Los parámetros a enviar para poder firmar un XML son los siguientes:

w= signXML

r= signFE

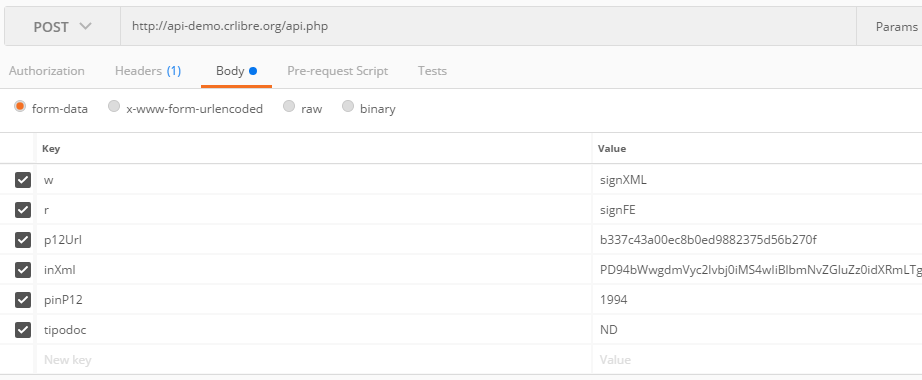
p12Url= Este es el código que se obtiene al subir el certificado

inXml= Este debe ser el XML que se va a firmar, pero debe ir en base64

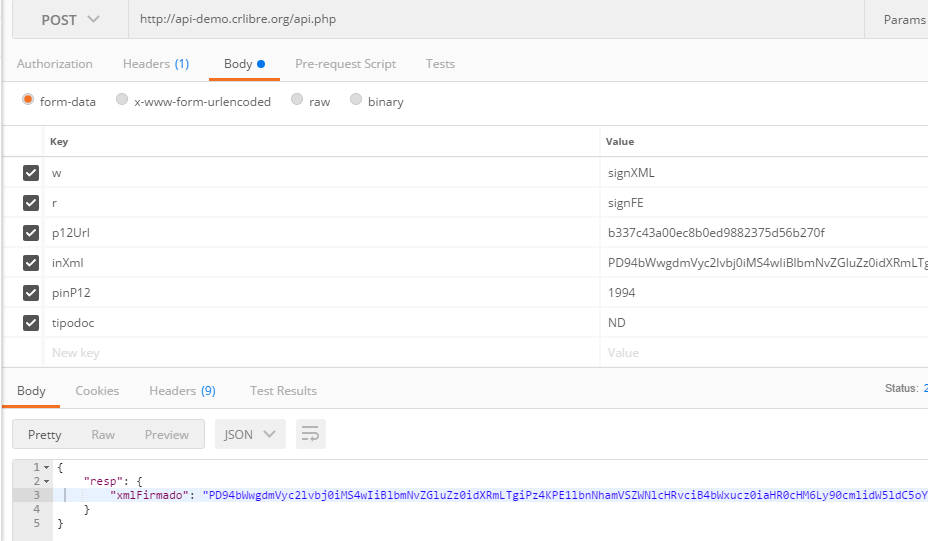
pinP12= Esta es la clave del certificado

tipodoc= Tipo de documento igual puede ser ND

b337c43a00ec8b0ed9882375d56b270f es el código que obtuvimos al subir el certificado.



El XML firmado viene en Base64, esto para que el parámetro se pase de una vez a envio a hacienda, la respuesta seria la siguiente:



### Firmado del XML Mensaje de Aceptación (Pendiente solucionar error)

Para firmar un XML se debe de hacer uso del Token del certificado, el cual subimos en la sección de [Upload del certificado o llave criptográfica](#_Upload_del_certificado)

Los parámetros a enviar para poder firmar un XML son los siguientes:

w= signXML

r= signFE

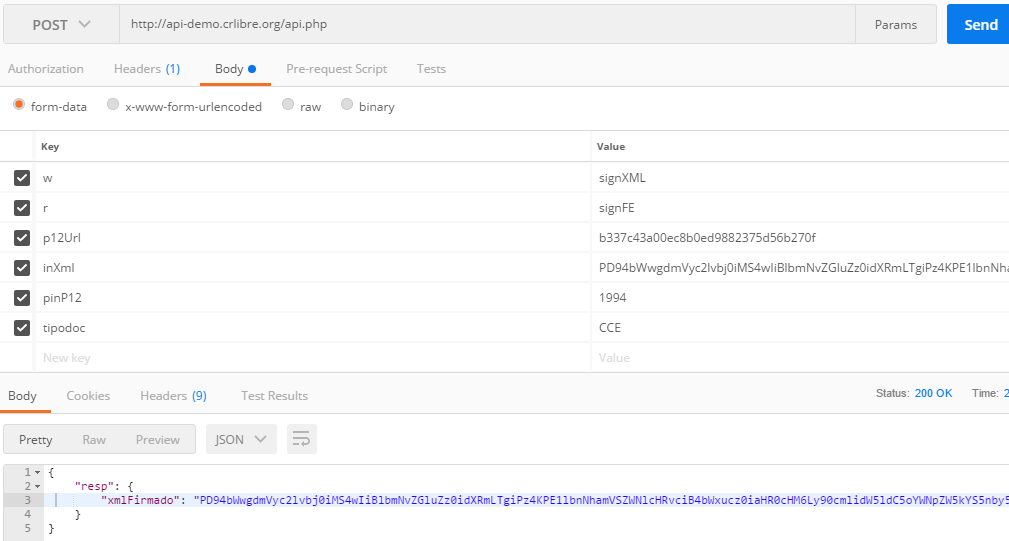
p12Url= Este es el código que se obtiene al subir el certificado

inXml= Este debe ser el XML que se va a firmar, pero debe ir en base64

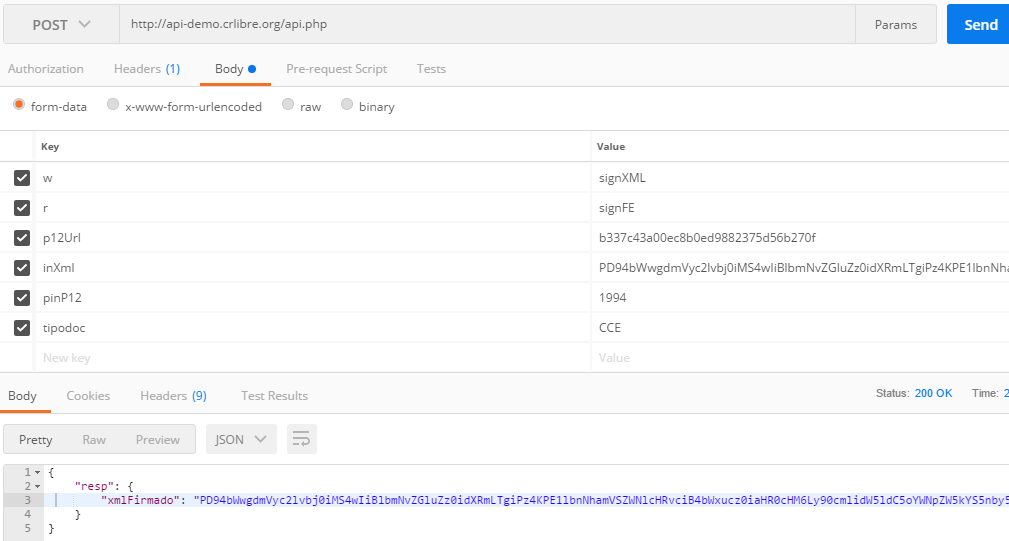
pinP12= Esta es la clave del certificado

tipodoc= Tipo de documento igual puede ser CCE CPCE RCE

b337c43a00ec8b0ed9882375d56b270f es el código que obtuvimos al subir el certificado



El XML firmado viene en Base64, esto para que el parámetro se pase de una vez a envio a hacienda, la respuesta seria la siguiente:



### Envió a Hacienda del XML de Factura Electrónica, Notas de Credito, Notas de Debito

Para realizar un envio de XML debo enviar al API los siguientes campos:

w:send

r:json

token: el token debe solicitarse antes, no puede ser vencido

clave: Esta es la clave del XML que vamos a enviar

fecha: La fecha debe ser la del XML

emi\_tipoIdentificacion: Esto es el tipo de cedula del emisor 01 físico o 02 jurídico

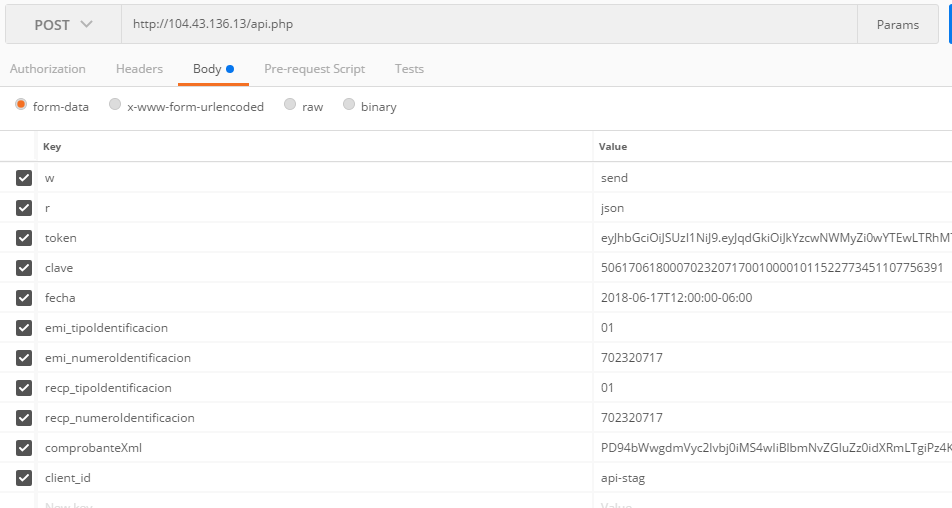
emi\_numeroIdentificacion: Este el numero de cedula del emisor

recp\_tipoIdentificacion: Esto es el tipo de cedula del emisor 01 físico o 02 jurídico

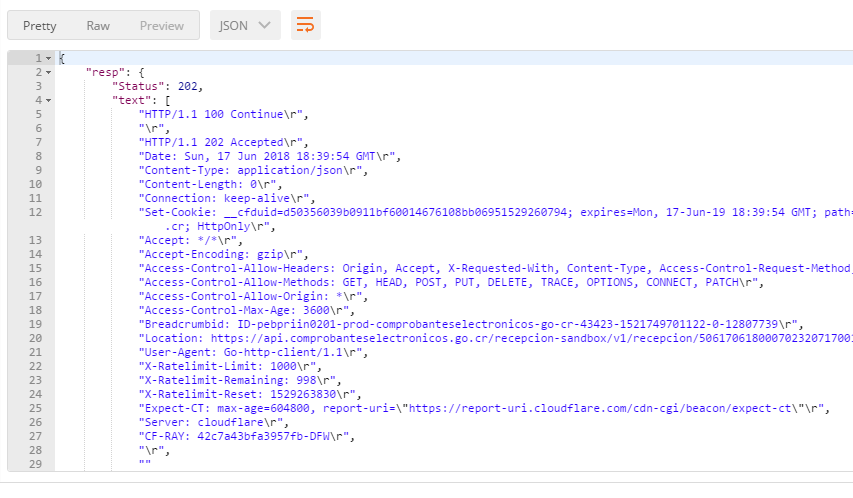
recp\_numeroIdentificacion: Este el numero de cedula del emisor

comprobanteXml: Este es el XML firmado y en base64

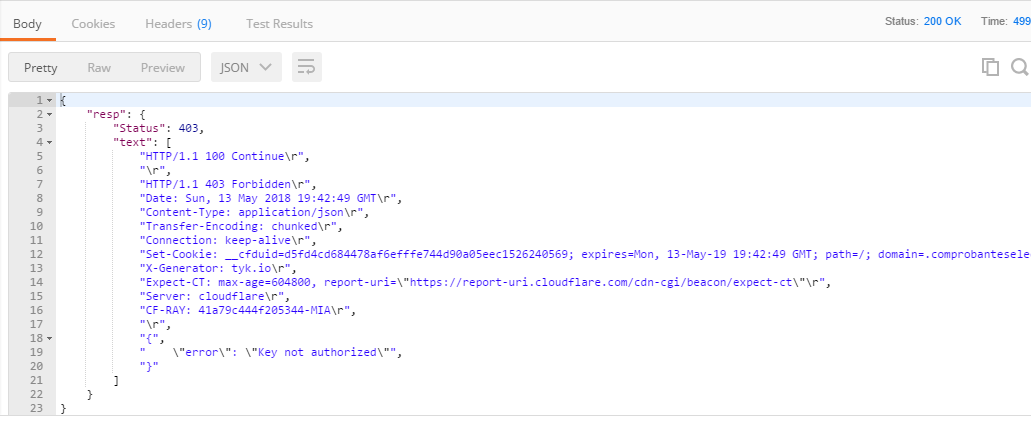
client\_id= api-stag o api-prod



La respuesta de Hacienda debe ser un 202, esto quiere decir que se recibió el XML, pero no explícitamente que fue aceptado, para verificar si fue aceptado debe realizar otra consulta.



En caso de no ser así hacienda responde un 403, la descripción del error lo pueden encontrar en text, ahí se describe la razón del error, normalmente es por un token ya vencido o por que la clave ya fue recibida, en este ejemplo dice al final error: Key not authorized eso es porque mi token ya se vencio.



### Envió a Hacienda del XML de Mensaje Aceptación (Aceptación total, Parcialmente y Rechazo)

### Comprobación de documentos

Para la comprobación de documentos se usa el modulo de consultar y se usa el método de consultarCom.

Funciona para consultar cualquiera de los tipos de documentos.

Los parámetros a enviar son los siguientes:

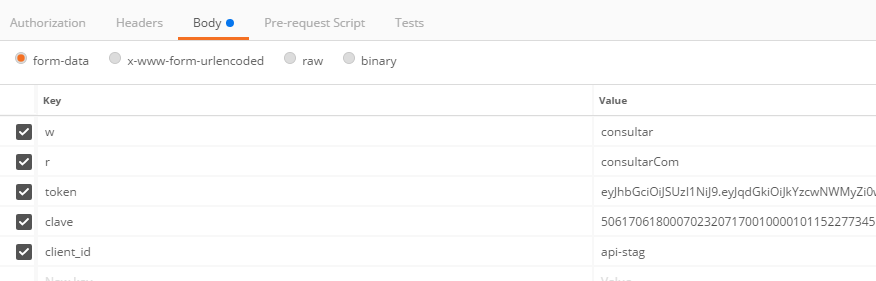
w:consultar

r:consultarCom

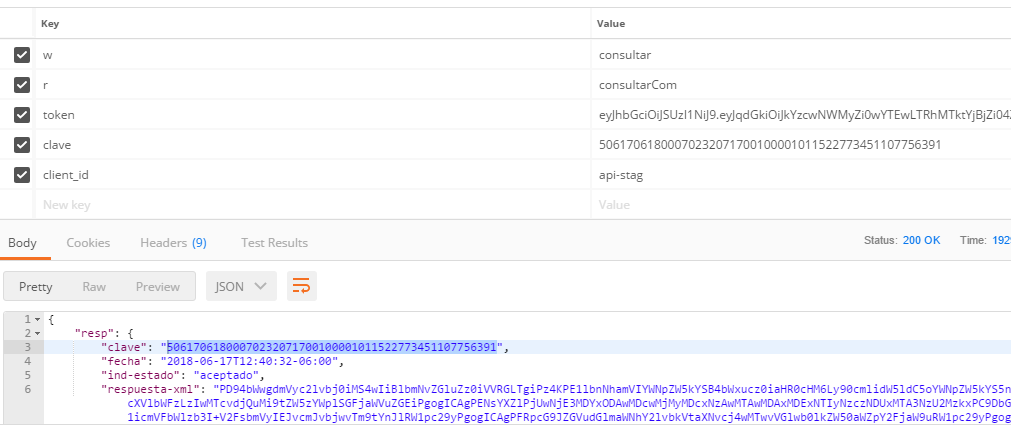
clave: La clave del documento a consultar

token: un token valido para hacer la consulta

client\_id: api-stag o api-prod



La respuesta nos va a devolver un ind-estado donde se refleja el estado del comprobante y además un respuesta-XML el cual viene en base64 y es obligatorio enviarlo al receptor jundo con el comprobante pdf y la factura XML. Por lo tanto, cuando se le envía al cliente la factura se deben enviar en total los 3 archivos.



## Info extra y modulos a corregir o pendientes de cambios:

### Información útil

Actualmente la mayoría de los módulos están en modo users\_openAccess, si alguien lo quiere poner para que sea users\_loggedIn lo puede hacer claro esta que debe de enviar el sessionKey.

### Pendientes:

* Revisión del proceso de mensaje de aceptación
* Agregar Documentación de los tiquetes electrónicos (Actualmente el Api maneja Tiquete Electrónico, pero no se encuentra en este documento)