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

Contenido

[CRLibre | Instrucciones de uso e instalación de API de Facturación electrónica 1](#_Toc513984942)

[Objetivo 2](#_Toc513984943)

[Módulos del API 2](#_Toc513984944)

[Instalación del Api 2](#_Toc513984945)

[Instalación de Base de Datos 2](#_Toc513984946)

[Instalación de los paquetes necesarios 2](#_Toc513984947)

[Primera petición al API 2](#_Toc513984948)

[Uso de Módulos del API 3](#_Toc513984949)

[Creación de Usuario 4](#_Toc513984950)

[Loging en el API 5](#_Toc513984951)

[Upload del certificado o llave criptográfica 6](#_Toc513984952)

[Solicitud de Token 7](#_Toc513984953)

[Solicitud de refrescar token 9](#_Toc513984954)

[Creación de Clave para los XML de Factura Electrónica 10](#_Toc513984955)

[Creación de Clave para Nota de Crédito 12](#_Toc513984956)

[Creación de Clave para Nota de Debito 13](#_Toc513984957)

[Creación de clave para Mensaje Aceptación (Aceptación total, Parcialmente y Rechazo) 15](#_Toc513984958)

[Creación de xml Factura Electrónica 17](#_Toc513984959)

[Creación de xml Nota de Crédito 17](#_Toc513984960)

[Creación de xml Nota de Debito 17](#_Toc513984961)

[Firmado del xml Factura Electrónica 17](#_Toc513984962)

[Firmado del xml Nota de Crédito 18](#_Toc513984963)

[Firmado del xml Nota de Debito 19](#_Toc513984964)

[Firmado del xml Mensaje de Aceptación (Pendiente solucionar error) 20](#_Toc513984965)

[Envió a Hacienda del xml de Factura Electrónica, Notas de Credito, Notas de Debito 21](#_Toc513984966)

[Envió a Hacienda del xml de Mensaje Aceptación (Aceptación total, Parcialmente y Rechazo) 23](#_Toc513984967)

[Comprobación de documentos 23](#_Toc513984968)

## 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.

## Módulos del API

* Creación de Usuarios
* Loging 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 Debito
* 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)
* Envió a Hacienda del xml de Factura Electrónica, Notas de Credito, Notas de Debito
* Envió a Hacienda del xml de Mensaje Aceptación (Aceptación total, Parcialmente y Rechazo)
* Consulta de estado de los comprobantes

# Instalación del Api

## Instalación de Base de Datos

## Instalación de los paquetes necesarios

## Primera petición al API

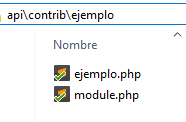
Las Peticiones al Api se realizan en Base al url o IP donde se encuentra el sitio, para mis efectos yo utilizare el API que se encuentra en demo de CRLibre.

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

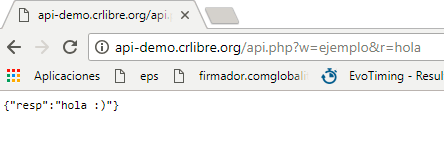


Para realizer este test vamos a enviarle al api 2 parametros, un W que es el nombre del modulo (Mejor dicho el nombre de la carpeta) y un R que es el método que ejecutara dentro del module.php

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 :)



## 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 utilizare PostMan para mostrar el funcionamiento de los diferentes Módulos.

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

Los otros modulos son meramente del Core del API.

Usare el modulo de Clave para mostrarle el uso y la estructura del modulo.

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

En el siguiente codigo veremos que $paths es el Array con las funciones, cuando hacemos un GET enviamos un w este es el nombre del modulo o la carpeta, ahora vemos en $paths un r' => 'hola', esto quiere decir que si yo quiero llegar a usar ese modulo debo enviar un w=clave y un r=ejemplo el 'action' => ' module\_hola ', quiere decir que ejecutara 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 unUsuario 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

### 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

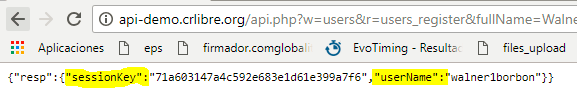
about = Esta es una informacion extra del usuario

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 como se loguea un usuario en el API para asi 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:

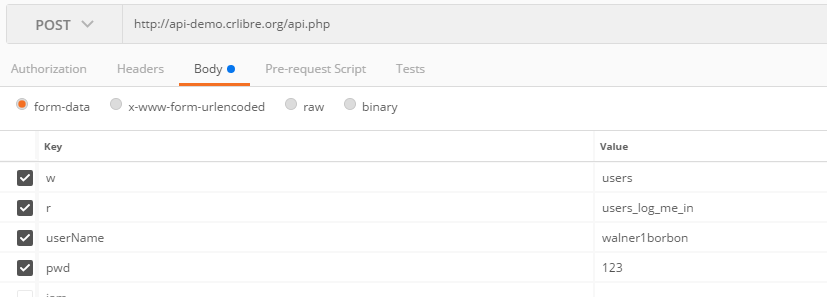
w = users

r = users\_log\_me\_in

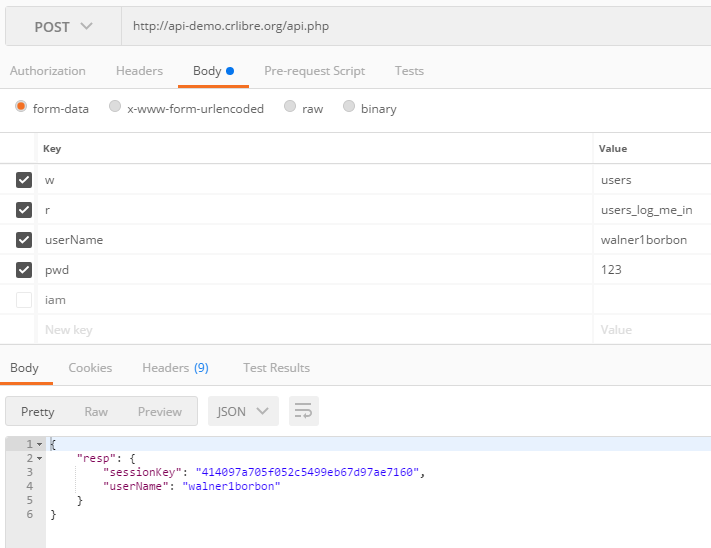
userName = userName previamente registrado

pwd= un password registrado

Parametros 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

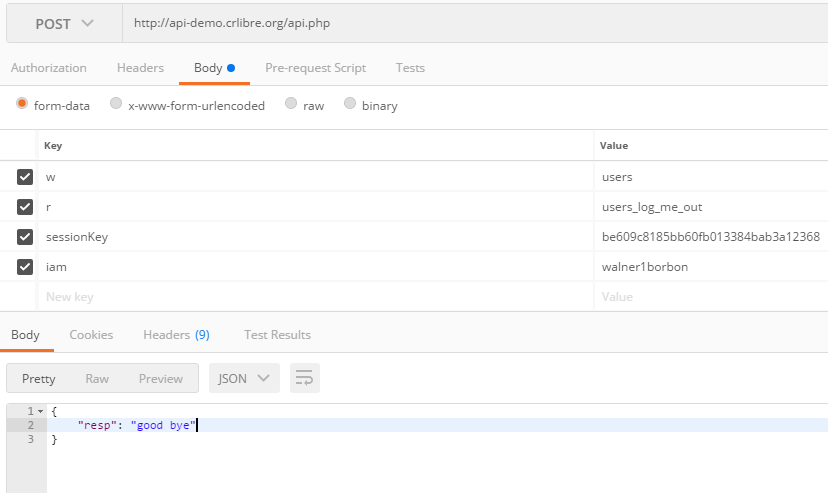
El api tiene la función de users\_log\_me\_out para cerrar la sesión, los parámetros a enviar son:

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:

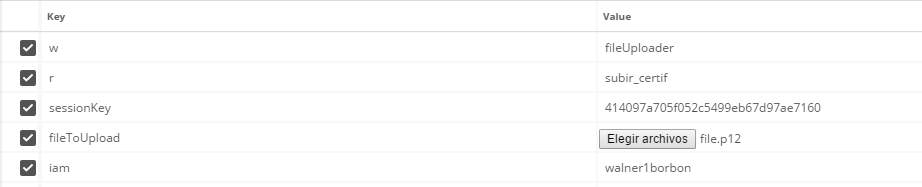
w = fileUploader

r = subir\_certif

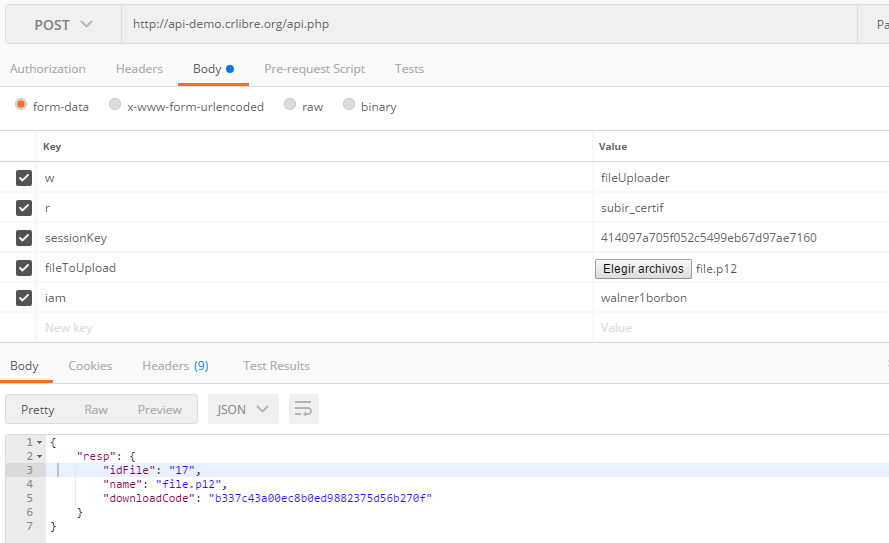
sessionKey= Esta la obtenemos previamente

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

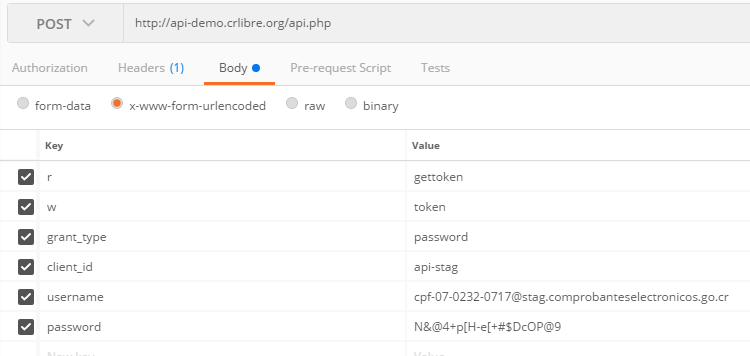
grant\_type = password para solicitar un token nuevo, mas adelante refresh\_token

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

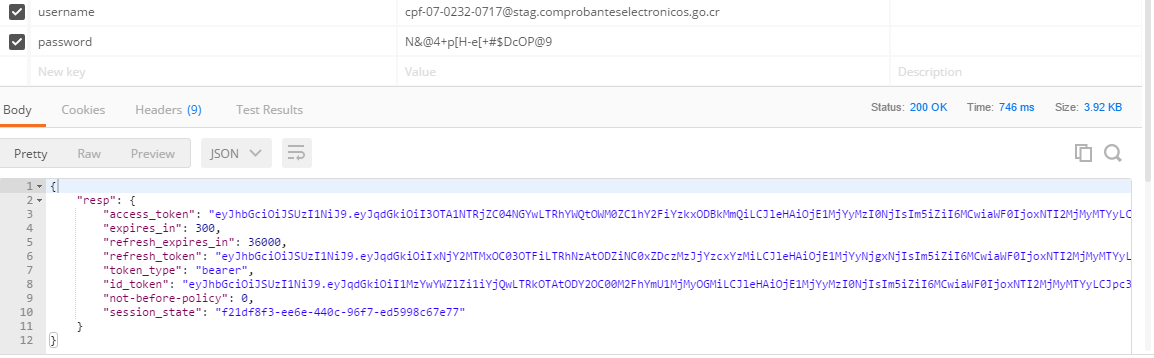
username= usuario de hacienda sambox o produccion

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

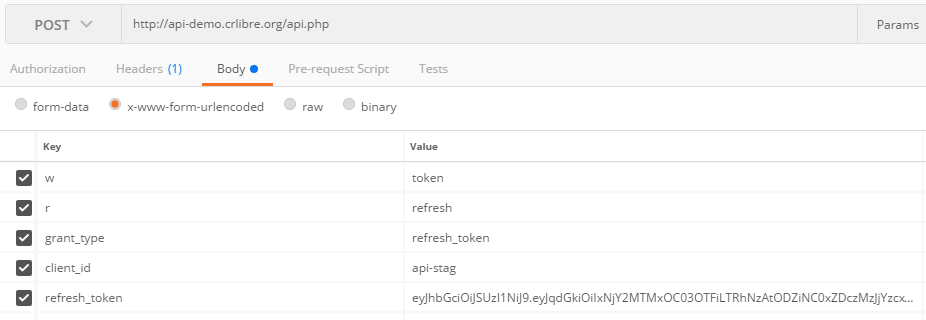
grant\_type= refresh\_token Esto dunciona solo 36000s después de solicitar el token

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

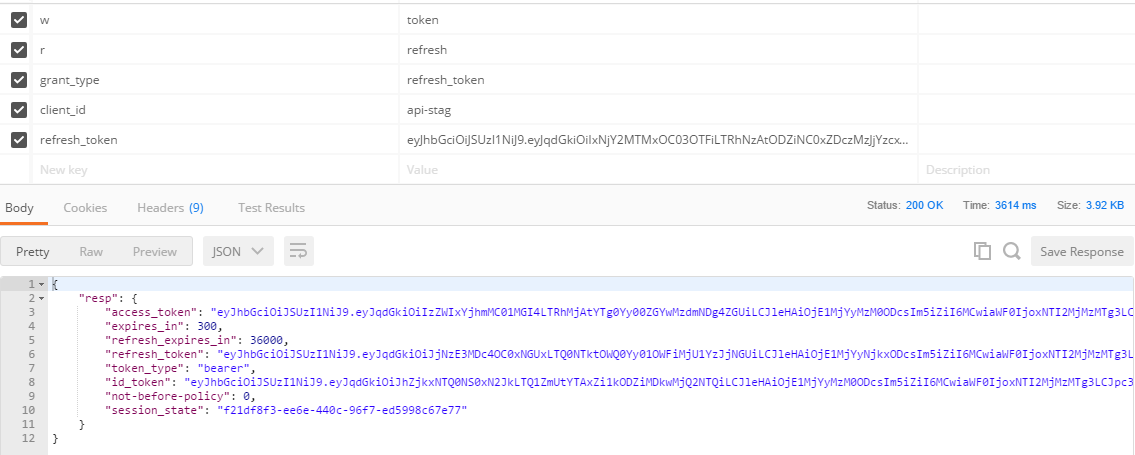
refresh\_token= el refresh\_token viene en la respuesta de solicitud del primer token



Por lo tanto la solicitud seria la siguiente



Y el resultado seria 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 contrucciones de claves.

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

tipoDocumento tiene las siguientes opciones:

FE Factura Electronica

ND Nota de Debito

NC Nota de Credito

TE Tiquete Electronico

CCE Confirmacion Comprabante Electronico

CPCE Confirmacion Parcial Comprbante Electronico

RCE Rechazo Comprobante Electronico

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

tipoCedula= fisico o juridico

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

situacion=

nomal En el caso de que la situación se nomal

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 paramentros (En mi caso Cedula Fisica)

w=clave

r=getClave

tipoCedula= fisico o juridico fisico  
 cedula= Numero de Cedula 702320717

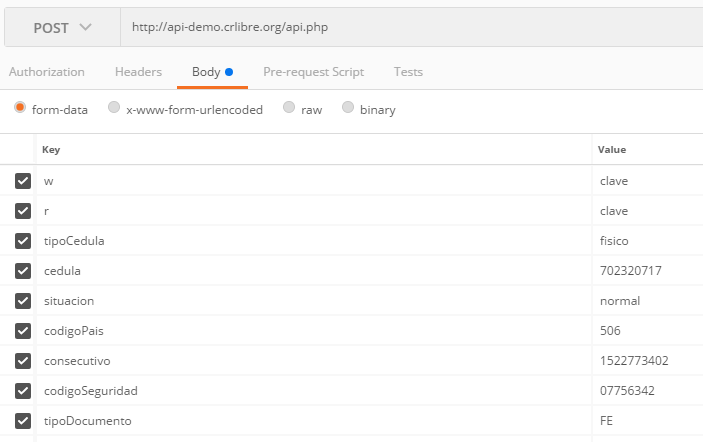
codigoPais= código del país 506

consecutivo= codigo de 10 numeros 1522773402

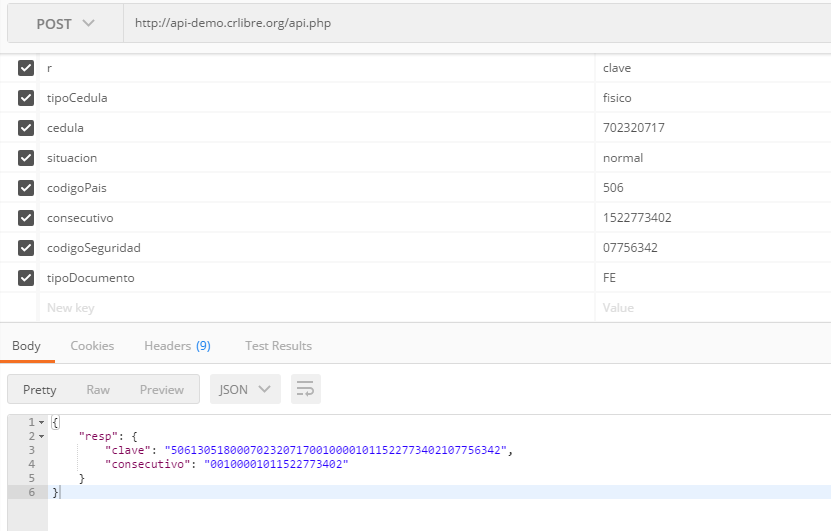
situacion= nomal contingencia sininternet normal

codigoSeguridad= codigo de 8 numeros 07756342

tipoDocumento= FE ND NC TE CCE CPCE RCE FE



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 Credito se envían los siguientes paramentros (En mi caso Cedula Fisica)

w=clave

r=getClave

tipoCedula= fisico o juridico fisico  
 cedula= Numero de Cedula 702320717

codigoPais= código del país 506

consecutivo= codigo de 10 numeros 1522773402

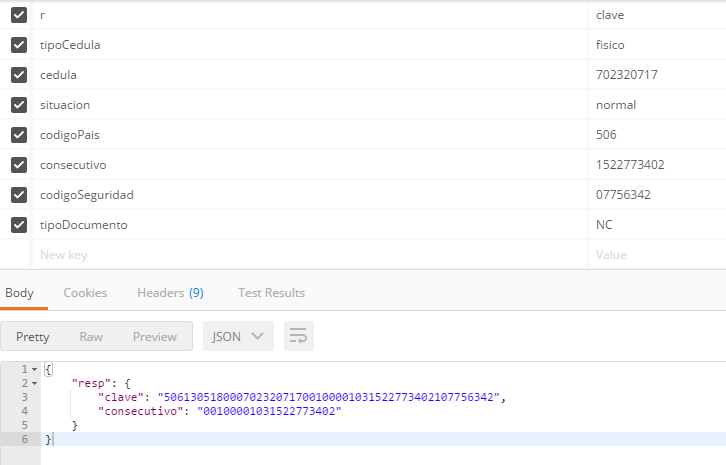
situacion= nomal contingencia sininternet normal

codigoSeguridad= codigo de 8 numeros 07756342

tipoDocumento= FE ND NC TE CCE CPCE RCE NC



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=getClave

tipoCedula= fisico o juridico fisico  
 cedula= Numero de Cedula 702320717

codigoPais= código del país 506

consecutivo= codigo de 10 numeros 1522773402

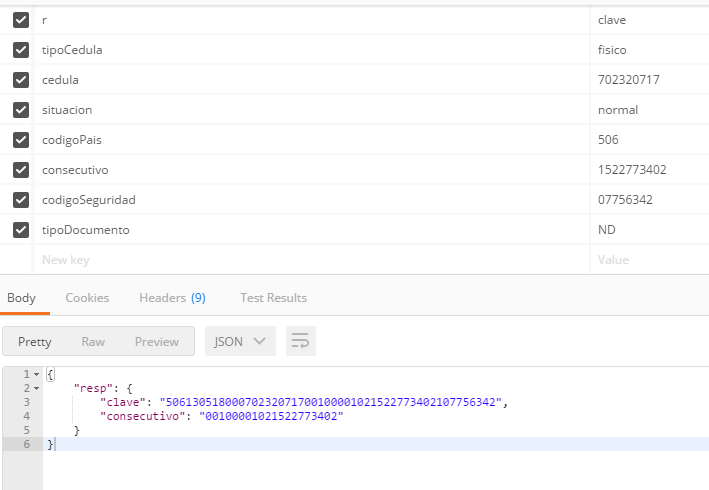
situacion= nomal contingencia sininternet normal

codigoSeguridad= codigo de 8 numeros 07756342

tipoDocumento= FE ND NC TE CCE CPCE RCE ND



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=getClave

tipoCedula= fisico o juridico fisico  
 cedula= Numero de Cedula 702320717

codigoPais= código del país 506

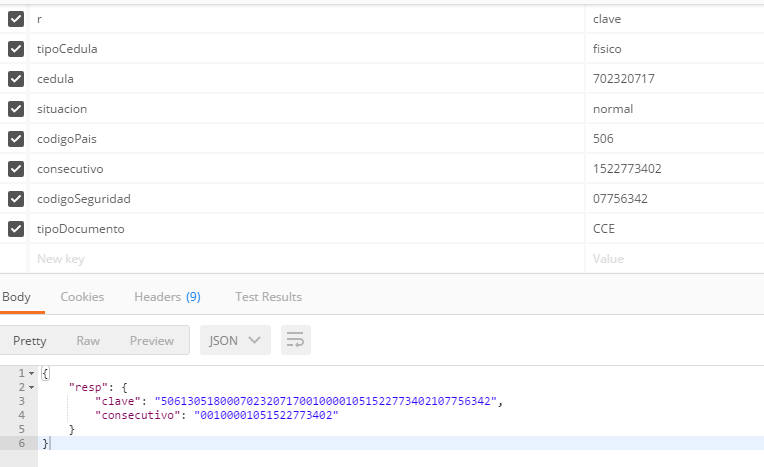
consecutivo= codigo de 10 numeros 1522773402

situacion= nomal contingencia sininternet normal

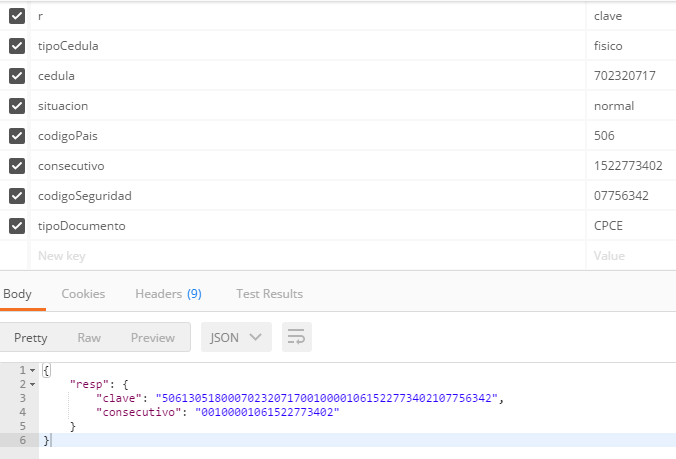
codigoSeguridad= codigo de 8 numeros 07756342

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

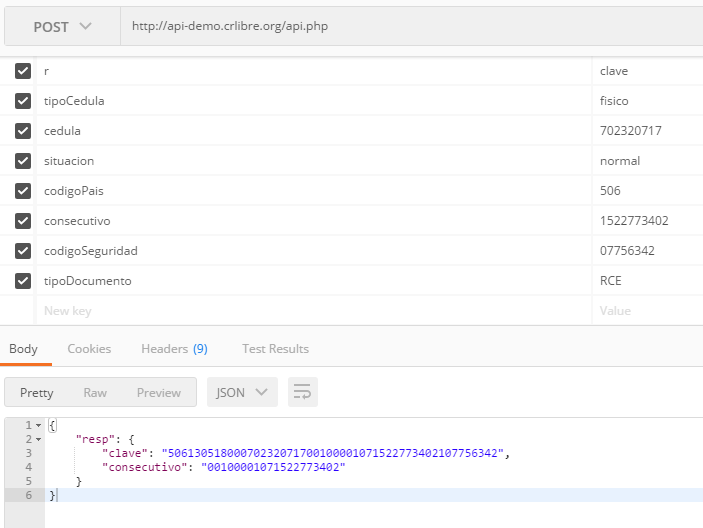
CCE | Confirmacion Comprabante Electronico



CPCE | Confirmacion Parcial Comprbante Electronico



RCE | Rechazo Comprobante Electronico



### 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 asi, que no omitan ninguno.

Los Datos son:

w:genXML

r:gen\_xml\_fe

clave:50613051800070232071700100001011522773408107756348

consecutivo:00100001011522773408

fecha\_emision:2018-05-13T15:30:00-06:00

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:Julian Subiros

receptor\_tipo\_identif:01

receptor\_num\_identif:114480790

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:julisubiros@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:200000

total\_merc\_gravada:0

total\_merc\_exenta:0

total\_gravados:0

total\_exentos:200000

total\_ventas:200000

total\_descuentos:0

total\_ventas\_neta:200000

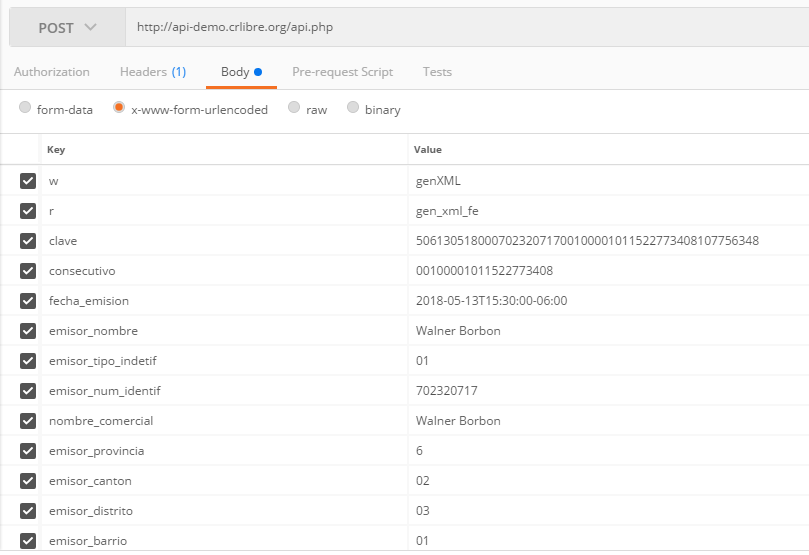
total\_impuestos:0

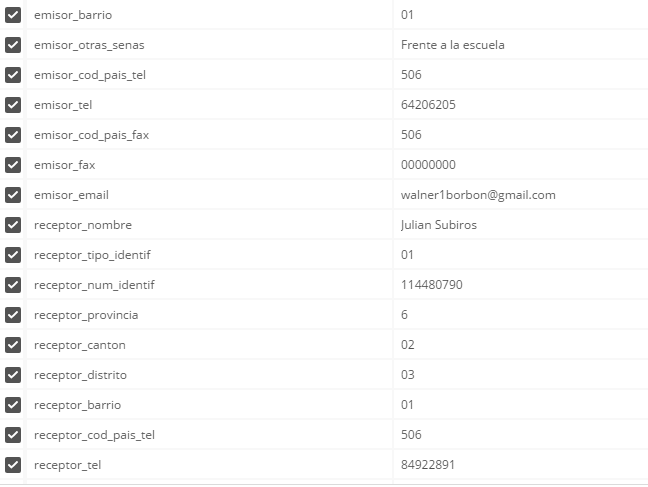
total\_comprobante:200000

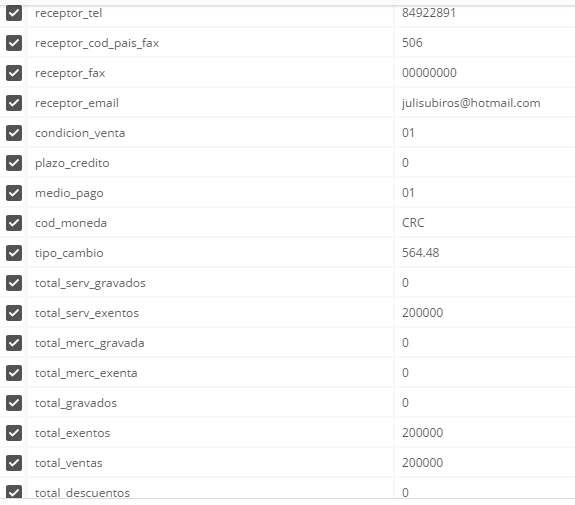
otros:Muchas gracias

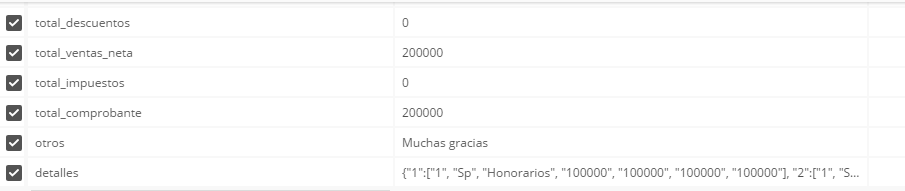
detalles:{"1":["1", "Sp", "Honorarios", "100000", "100000", "100000", "100000"], "2":["1", "Sp", "Honorarios", "100000", "100000", "100000", "100000"]}

El campo detalle debe enviarse con la siguiente estructura detalles:{" NumeroLinea":["Cantidad", "Sp", "Detalle", "PrecioUnitario", "MontoTotal", "SubTotal", "MontoTotalLinea"]}



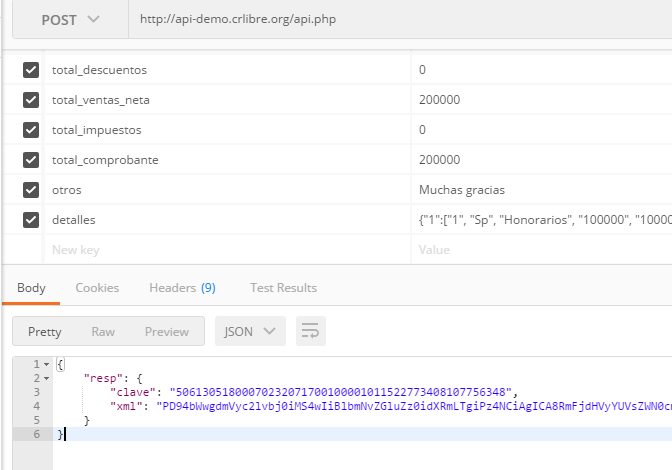






La respuesta del API será clave y xml, pero ese xml esta sin firmar y en base64

Por lo que deberá luego ser firmado.



### Creación de xml Nota de Crédito

### Creación de xml Nota de Debito

### 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

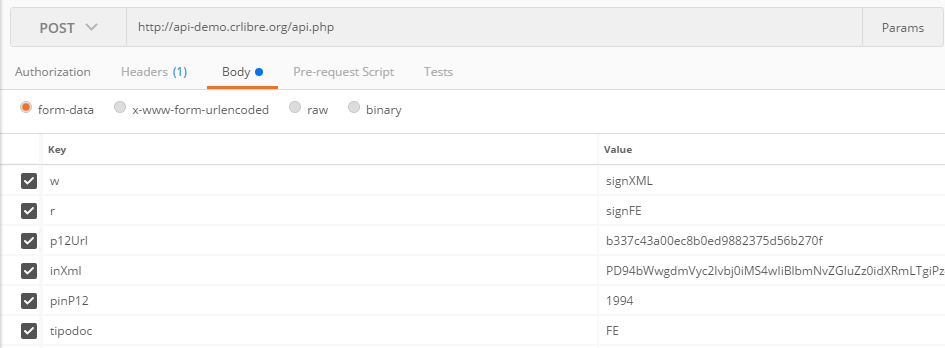
p12Url= Este es el código que se obtiene al subir el certificado

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

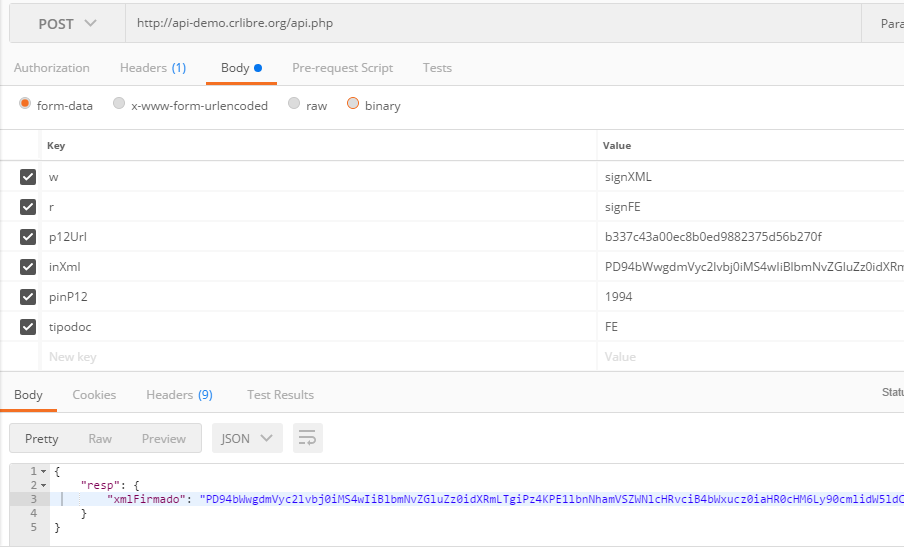
pinP12= Esta es la clave del certificado

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

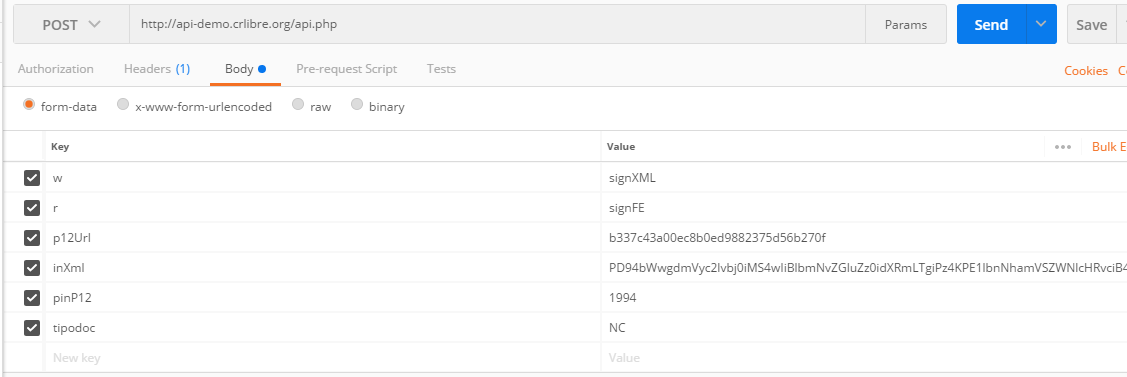
p12Url= Este es el código que se obtiene al subir el certificado

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

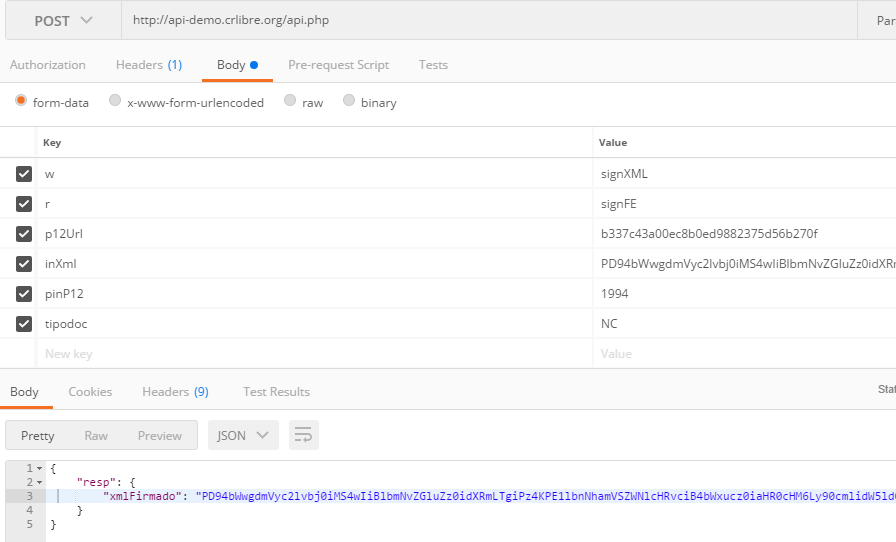
pinP12= Esta es la clave del certificado

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

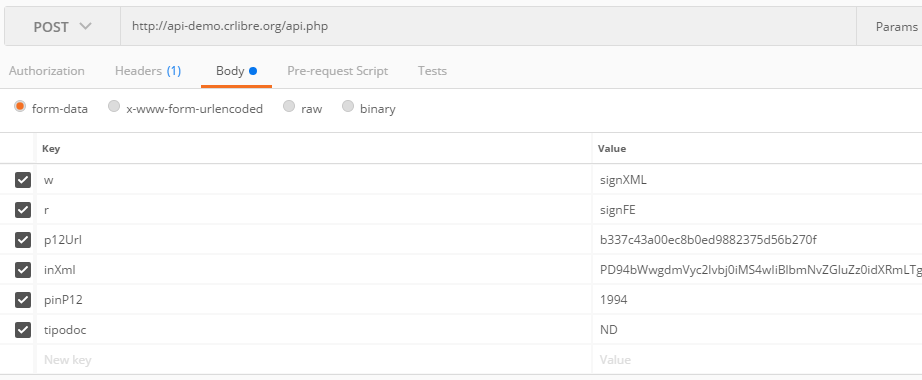
p12Url= Este es el código que se obtiene al subir el certificado

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

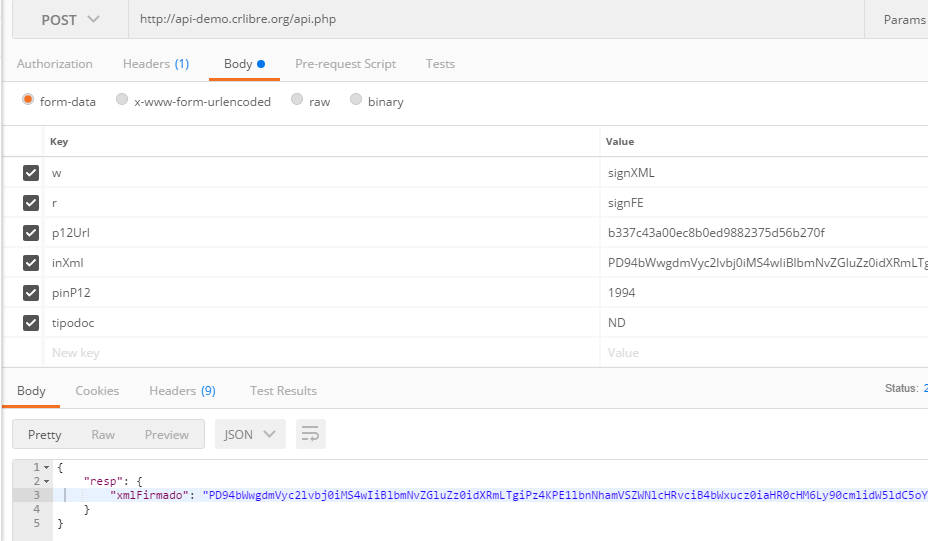
pinP12= Esta es la clave del certificado

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

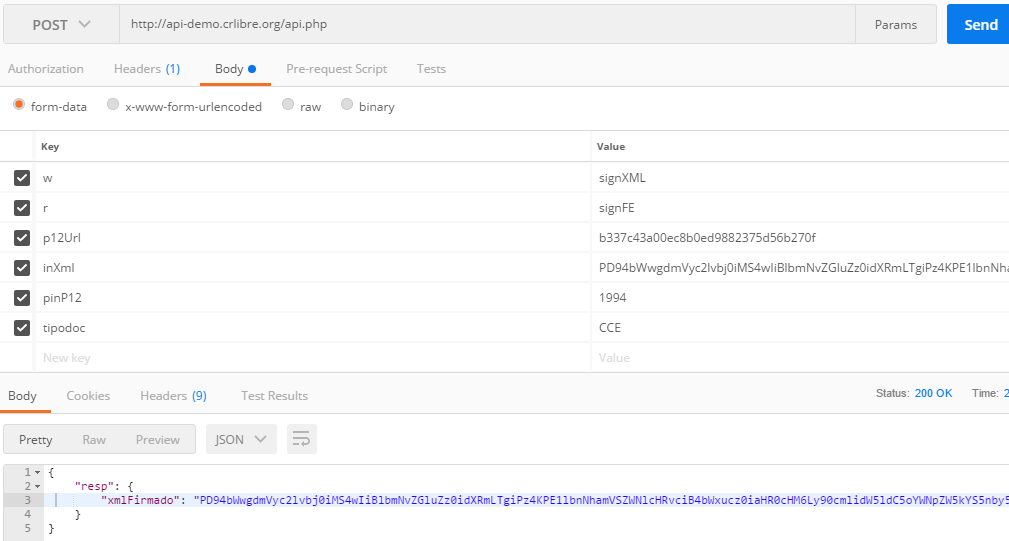
p12Url= Este es el código que se obtiene al subir el certificado

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

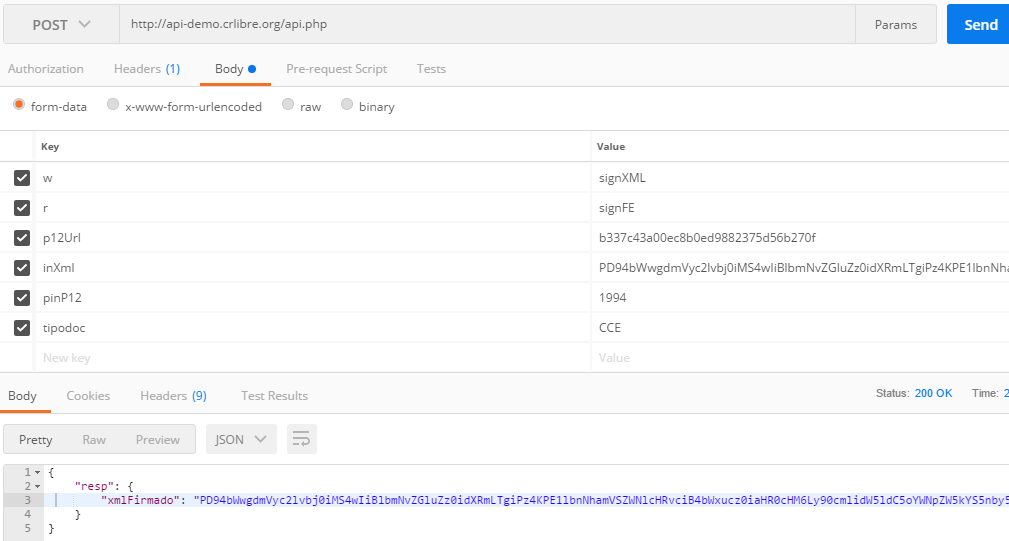
pinP12= Esta es la clave del certificado

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

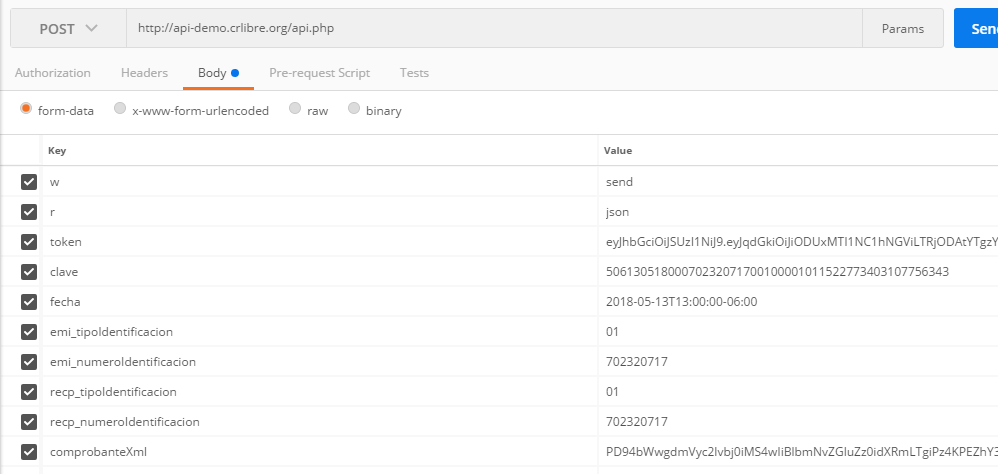
emi\_tipoIdentificacion: Esto es el tipo de cedula del emisor 01 fisico o 02 juridico

emi\_numeroIdentificacion: Este el numero de cedula del emisor

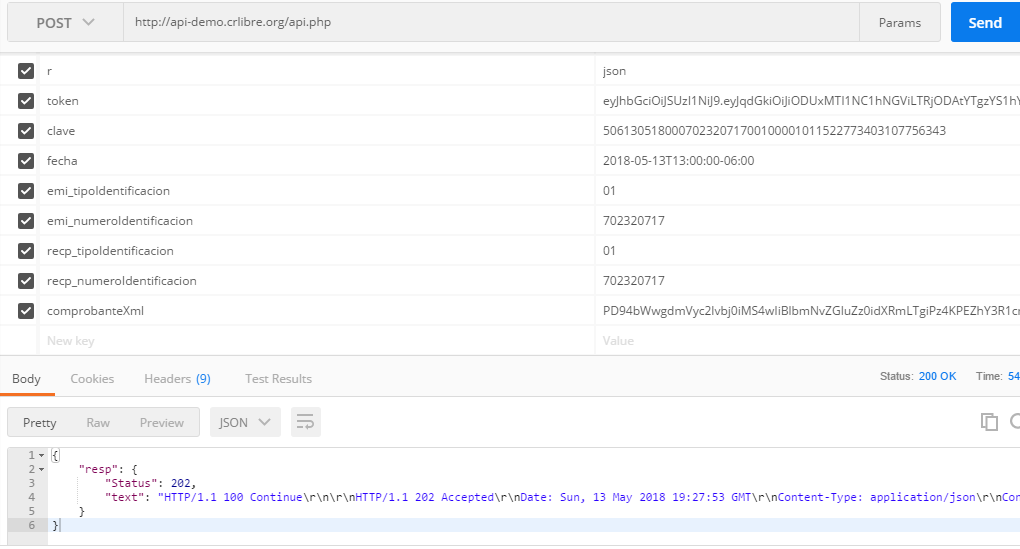
recp\_tipoIdentificacion: Esto es el tipo de cedula del emisor 01 fisico o 02 juridico

recp\_numeroIdentificacion: Este el numero de cedula del emisor

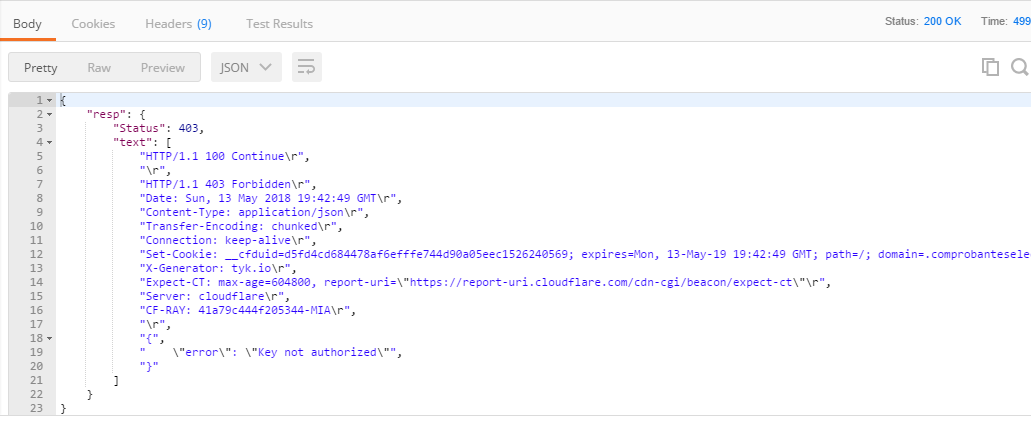
comprobanteXml: Este es el xml firmado y en base64



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 asi 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.

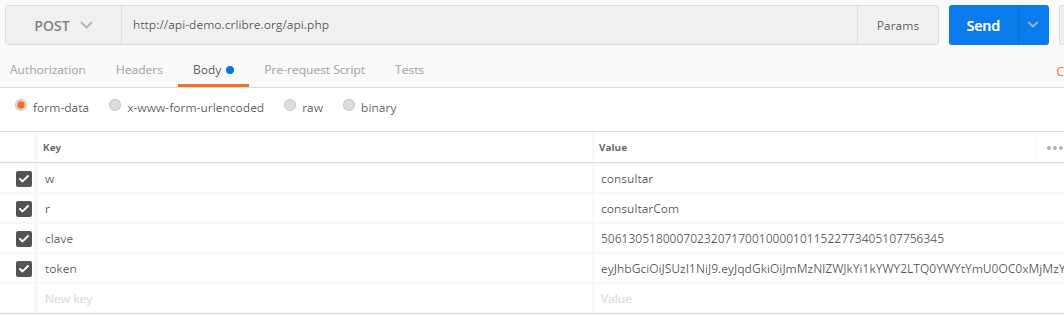
Los parámetros a enviar son los siguientes:

w:consultar

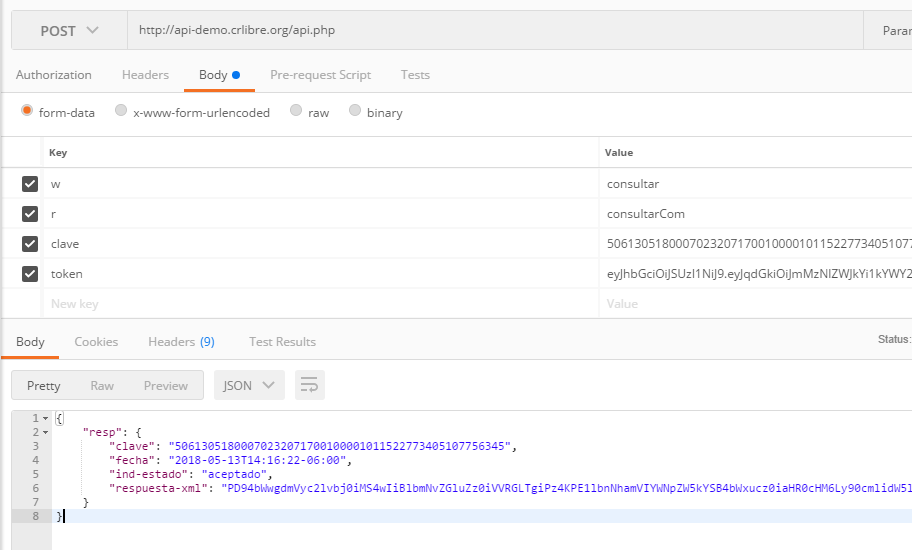
r:consultarCom

clave: La clave del documento a consultar

token: un token valido para hacer la consulta



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.



## Pendientes de Documentacion:

### [Instalación de Base de Datos](#_Instalación_de_Base)

### [Instalación de los paquetes necesarios](#_Instalación_de_los)

### [Creación de Factura Electronica](#_Creación_de_Clave)

### [Creación de xml Nota de Credito](#_Creación_de_Clave_1)

### [Creación de xml Nota de Debito](#_Creación_de_xml)

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

Creación de los XML de ND NC Mensaje aceptacion( Ya se esta trabajando en eso)

Actualmente la mayoría de los modulos 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.

El api tiene un tiempo de sesión de 3600s (ubicado en modulo de users archivo module.php línea 462) eso lo pueden cambiar para que sea menor