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

Contenido

[CRLibre | Instrucciones de uso e instalación de API de Facturación electrónica 1](#_Toc513837335)

[Objetivo 2](#_Toc513837336)

[Módulos del API 2](#_Toc513837337)

[Instalación del Api 2](#_Toc513837338)

[Instalación de Base de Datos 2](#_Toc513837339)

[Instalación de los paquetes necesarios 2](#_Toc513837340)

[Primera petición al API 2](#_Toc513837341)

[Uso de Módulos del API 2](#_Toc513837342)

[Creación de Usuario 3](#_Toc513837343)

[Loging en el API 3](#_Toc513837344)

[Upload del certificado o llave criptográfica 3](#_Toc513837345)

[Solicitud de Token 3](#_Toc513837346)

[Solicitud de refrescar token 3](#_Toc513837347)

[Creación de Clave para los XML de Factura Electrónica 3](#_Toc513837348)

[Creación de Clave para Nota de Crédito 3](#_Toc513837349)

[Creación de Clave para Nota de Debito 3](#_Toc513837350)

[Creación de clave para Mensaje Aceptación (Aceptación total, Parcialmente y Rechazo) 3](#_Toc513837351)

[Creación de xml Factura Electrónica 3](#_Toc513837352)

[Creación de xml Nota de Crédito 3](#_Toc513837353)

[Creación de xml Nota de Debito 3](#_Toc513837354)

[Firmado del xml Factura Electrónica 3](#_Toc513837355)

[Firmado del xml Nota de Crédito 3](#_Toc513837356)

[Firmado del xml Nota de Debito 3](#_Toc513837357)

[Firmado del xml Mensaje de Aceptación (Pendiente solucionar error) 3](#_Toc513837358)

[Envió a Hacienda del xml de Factura Electrónica 3](#_Toc513837359)

[Envió a Hacienda del xml de Nota de Debito 3](#_Toc513837360)

[Envió a Hacienda del xml de Nota de Crédito 3](#_Toc513837361)

[Envió a Hacienda del xml de Mensaje Aceptación (Aceptación total, Parcialmente y Rechazo) 3](#_Toc513837362)

## 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
* Envió a Hacienda del xml de Nota de Debito
* Envió a Hacienda del xml de Nota de Crédito
* Envió 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 del Api

## Instalación de Base de Datos

## Instalación de los paquetes necesarios

## Primera petición al API

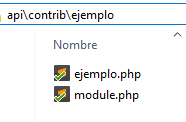
Las Peticiones al Api se realizan en Base al url o IP donde se encuentra el sitio, para mis efectos yo utilizare el API que se encuentra en demo de CRLibre.

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

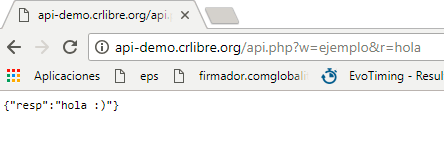


Para realizer este test vamos a enviarle al api 2 parametros, un W que es el nombre del modulo (Mejor dicho el nombre de la carpeta) y un R que es el método que ejecutara dentro del module.php

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

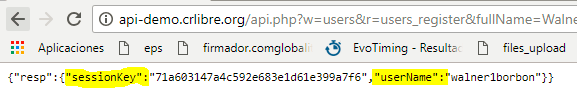
about = Esta es una informacion extra del usuario

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:

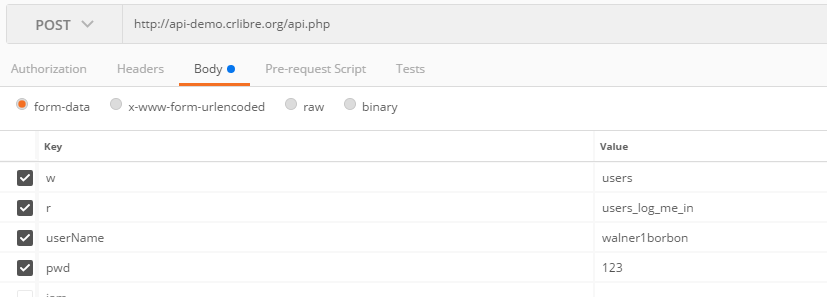
w = users

r = users\_log\_me\_in

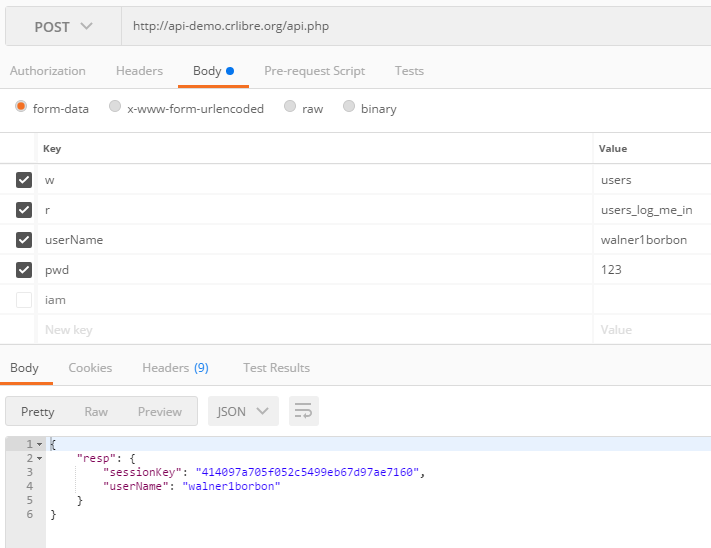
userName = userName previamente registrado

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

### 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:

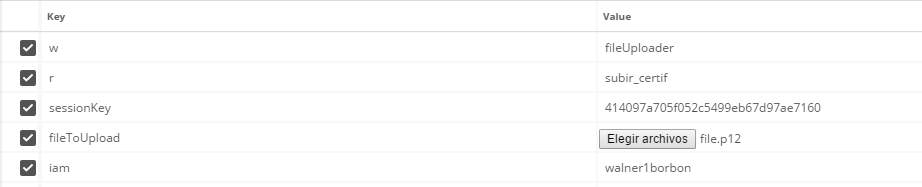
w = fileUploader

r = subir\_certif

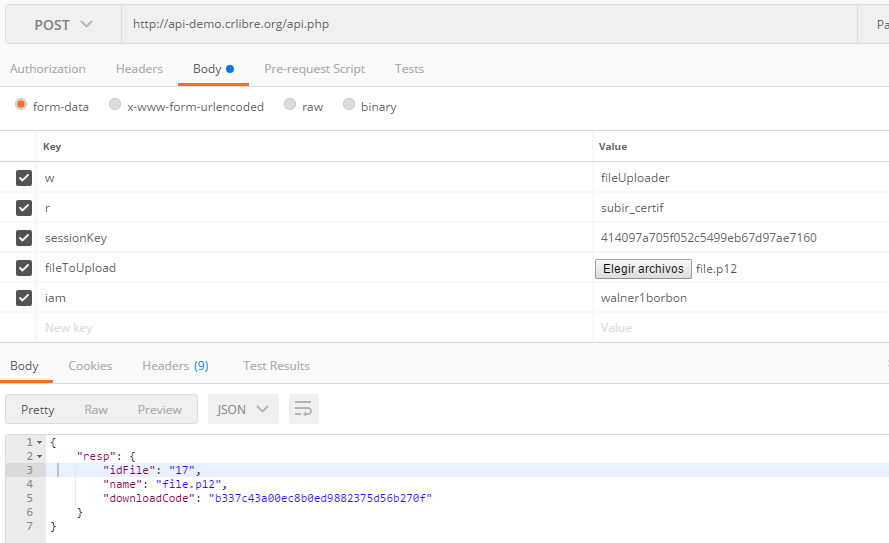
sessionKey= Esta la obtenemos previamente

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

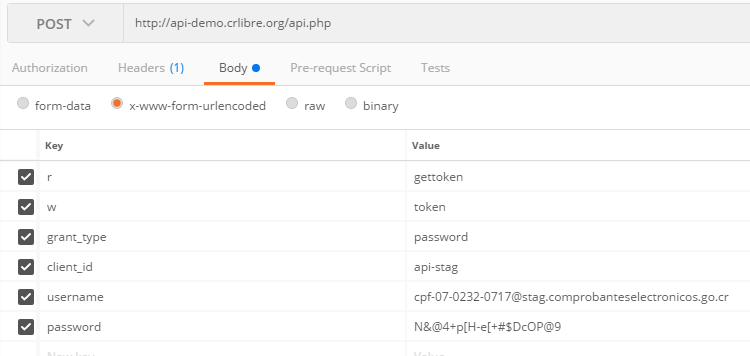
grant\_type = password para solicitar un token nuevo, mas adelante refresh\_token

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

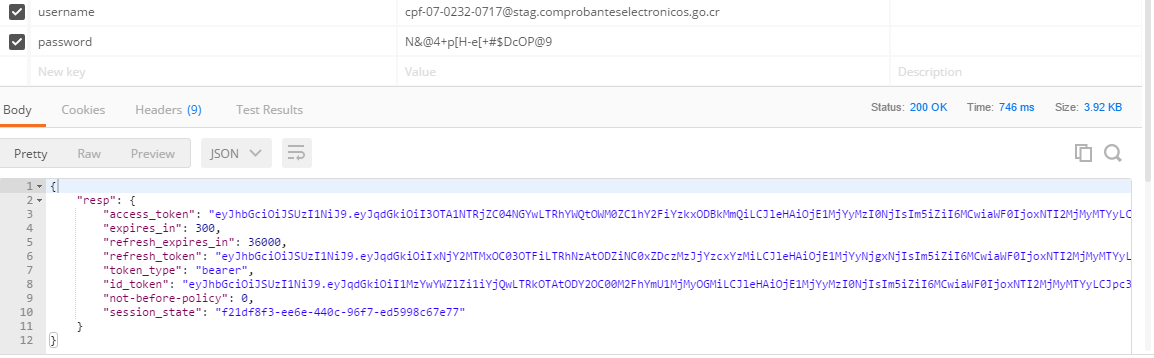
username= usuario de hacienda sambox o produccion

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

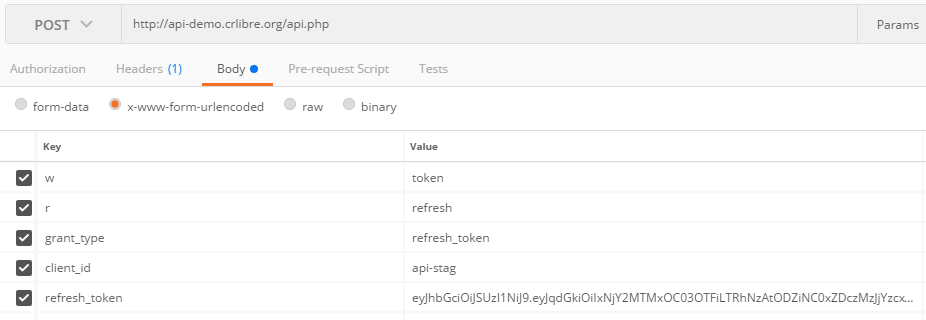
grant\_type= refresh\_token Esto dunciona solo 36000s después de solicitar el token

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

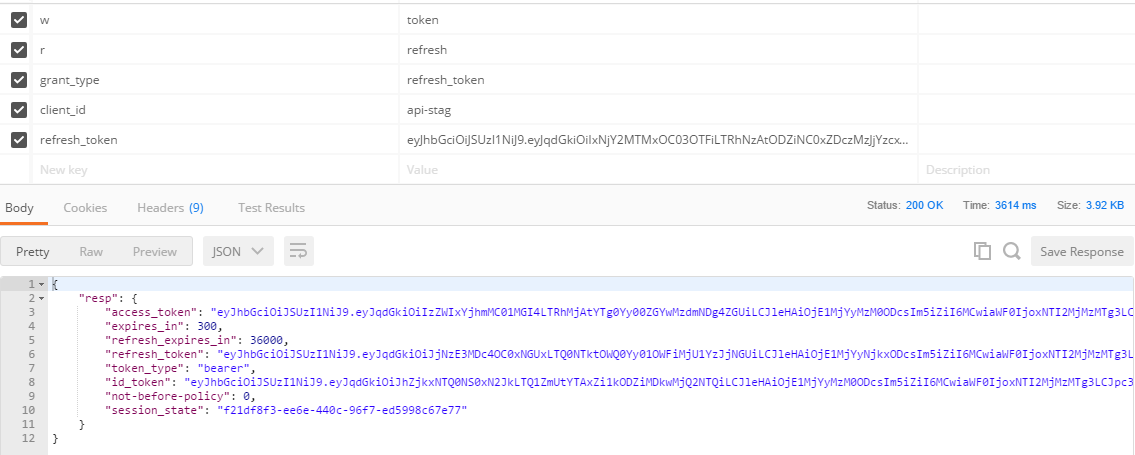
refresh\_token= el refresh\_token viene en la respuesta de solicitud del primer token



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

### 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

### Envió a Hacienda del xml de Nota de Debito

### Envió a Hacienda del xml de Nota de Crédito

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