Manual de Usuario

API Timbrado Factorum

Objetivo

El nuevo servicio de timbrado tiene como objetivo mejorar el proceso de timbrado, además se añadieron opciones para emisiones masivas en los servicios de emisión y emisión de prueba.

Accesos

En todos los endpoints para solicitar el timbrado es necesario proporcionar las credenciales (correo, RFC y password) de la cuenta contratada con Factorum.

Contratos Firmados y Vigentes

Los contratos de prestación del servicio deben estar firmados y vigentes a la fecha del uso de cada petición.

Créditos de la cuenta

La cuenta debe contar con una fecha de vigencia y tener folios disponibles para realizar el timbrado de la petición.

Servicios

El servicio o API de Timbrado Factorum cuenta con métodos para realizar operaciones relacionadas al timbrado de uno o muchos documentos.

Al contrario del servicio anterior si se colocan las urls en un navegador no se mostrarán los metadatos de los métodos; el servicio solo se puede utilizar a través de la invocación de los métodos como se muestra en la sección de ejemplos.

Timbrado de un documento.

Para éste servicio, se enviará una petición con las credenciales de su cuenta Factorum y el xml, deberá enviarse en un json con todos los datos (en la sección de ejemplos se encuentra una muestra para el lenguaje c#).

Los campos de Usuario, password, rfc y xml son strings, el xml es un arreglo de byte o un string en base 64.
El header en la petición debe contener: "Content-Type: application/json"

La respuesta a las peticiones contiene los siguientes campos:

- ReturnStringXML - String
- ReturnFileXML - byte[]
- ReturnFileQRCode - byte[]
  - Es una imagen gif que se recibirá como arreglo de byte
- FechaTimbrado - DateTime
- UUID - String
- SelloCFD - String
- NoCertificadoSAT - String
- SelloSAT - String
- CadenaOriginalSello - String
- CadenaOriginalTimbre - String
- MensajeError - String
- Código - Integer
  - La lista de los códigos posibles se enlista en la sección **Códigos de Mensajes**

A continuación se detallan más los métodos a utilizar:

➔ FactorumGenYaSelladoConArchivo

Este método es el utilizado para timbrar un CFDI (xml) y al concluir satisfactoriamente devuelve el xml con el complemento de Timbre Fiscal con un Folio Fiscal válido ante el SAT.

Para éste se pueden utilizar dos métodos según sea el caso y las emisiones serán reales:

http://factorumweb.com/ApiTimbrado/Timbrado/FactorumGenYaSelladoConArchivo

http://factorumweb.com/ApiTimbrado/Timbrado/FactorumGenYaSelladoConArchivoBase64

➔ FactorumGenYaSelladoConArchivoTest

Este método es el utilizado para timbrar un CFDI (xml) pero en modo de prueba y de igual forma devolverá un xml timbrado con folio fiscal, pero este NO tiene validez ante el SAT.

http://factorumweb.com/ApiTimbrado/Timbrado/FactorumGenYaSelladoConArchivoTest/

http://factorumweb.com/ApiTimbrado/Timbrado/FactorumGenYaSelladoConArchivoTestBase64/

En la sección de ejemplos se muestra la llamada al servicio de timbrado con éste método de ejemplo, hay que tomar en cuenta que para algunos lenguajes sea necesario hacer un paso extra en algunos campos como el string del xml o los arreglos de byte tanto para xml como el QR. Los ejemplos están en lenguaje c#.

Para el archivo de QR está generado como archivo GIF.

Timbrado por volumen o masivo

Éste nuevo servicio está diseñado para emisores de volúmenes considerables de CFDIs, el cual evita que se tenga que realizar la petición n veces.

Consiste en hacer la petición con las credenciales de su cuenta Factorum, pero con la diferencia de mandar un zip con los CFIDs a emitir en vez de solo el CFID, éste le devolverá un id de petición que utilizará en los demás servicios para dar seguimiento.

Después de realizar la primera petición con la solicitud de todos los CFIDs que desea emitir, se deberá hacer la petición de estatus dentro de las siguientes 24hrs, se recomienda hacer la petición 2hr después de la primera.

La petición de Estatus mostrará si ya está disponible o en proceso la petición (los estatus se enlistan en el método correspondiente), si muestra que ya está lista se podrá realizar la siguiente petición.

El siguiente paso es el de descarga de los archivos, se realizará la petición (con el identificador proporcionado) se obtendrá el archivo zip con los archivos timbrados y un log.

A continuación se enlistan los métodos antes mencionados:

➔ Timbrado Masivo

Este método permite realizar una solicitud para timbrar más de 1 xml en una sola petición, será procesado dentro de las siguientes 24 Horas. La extensión del archivo que tendrá todos xmls por emitir deberá ser zip; la respuesta contendrá:

- ➢ IdSolicitud - String
- ➢ FechaSolicitud - DateTime
- ➢ Codigo - Integer
- ➢ Mensaje - String
- ➢ Estatus - String

El **IdSolicitud** es el que se utilizará para conocer el estatus de la petición y para solicitar los archivos.

http://factorumweb.com/ApiTimbrado/Timbrado/TimbradoMasivo/

➔ Timbrado Masivo Test

Realiza la misma tarea que la anterior, pero en modo de prueba por lo que los xml que se descarguen derivado de esta petición no tendrán validez ante el SAT. Genera un id de petición para el seguimiento del mismo.

http://factorumweb.com/ApiTimbrado/Timbrado/TimbradoMasivoTest/

➔ Obtener Estatus Timbrado Masivo

Este método sirve para monitorear el progreso de la solicitud de timbrado masivo, se medirá en Status **Pendiente, En Proceso, Finalizado, Error Interno**. Requiere el id de petición.

http://factorumweb.com/ApiTimbrado/Timbrado/GetStatusTimbradoMasivo/<idpeticion>

➔ Descargar Archivos

Este método realiza la descarga de los xml timbrados exitosamente de la petición indicada. Requiere el id de petición.

http://factorumweb.com/ApiTimbrado/Timbrado/DescargarArchivos/<idPeticion>

Códigos de Mensajes

Códigos Autenticación:

412: El usuario o RFC que está intentando utilizar no está disponible

600: No definido.

601: Usuario sin credenciales (No se proporcionó los datos para realizar la autenticación)

602: El usuario NO existe

603: Usuario inactivo.

604: Cuenta sin créditos disponible.

605: El crédito del usuario no tiene fecha de vencimiento establecida o vigente.

606: La cuenta agotó el límite de folios contratados.

607: El usuario no tiene vigente ningún plan de timbrado.

608: Ocurrió un error interno en la autenticación.

Códigos Timbrado:

100: No se emitió el documento.

101: Ocurrió un error que será validado con el SAT.

102: Se detectó un error de timeout, esto se puede deber a problemas en el SAT por lo que no quiere decir que no se haya emitido el CFDI

103: Timbrado previamente, se identificó que el CFDI ya se había mandado con anterioridad.

200: Documento firmado exitosamente o solicitud registrada.

201: Solicitud de documentos registrada.

501: Ocurrió un error al intentar procesar su solicitud.

Códigos Estatus y Descarga Timbrado Masivo:

204: El identificador de solicitud no es válido.

404: No existe el identificador de solicitud.

Ejemplos de peticiones

Se muestran algunos ejemplos para realizar las peticiones de los servicios de emisión:

FactorumGenYaSelladoConArchivoTest

```
String urlApi = "http://api cloud.factorumweb.com/ApiTimbrado/Timbrado/FactorumGenYaSelladoConArchivoTest/";
var @params = new Dictionary<string, object>();
@params.Add("Usuario", usuario);
@params.Add("Password", password);
@params.Add("RFC", rfc);
@params.Add("XML", xml);
JsonSerializerSettings settings = new JsonSerializerSettings();
settings.MaxDepth = int.MaxValue;
var inputJson = JsonConvert.SerializeObject(@params, settings);

var content = new StringContent(inputJson, System.Text.Encoding.UTF8, "application/json");

using (HttpClient cliente = new HttpClient())
{
    using (var response = await cliente.PostAsync(urlApi, content))
    {
        if (response.StatusCode != HttpStatusCode.OK)
        {
            return responseFactorum;
        }

        string apiResponse = await response.Content.ReadAsStringAsync();
        responseFactorum = JsonConvert.DeserializeObject<FactorumResponse>(apiResponse);
    }
}
return responseFactorum;
```

Ejemplo de Json para la petición de timbrado

```
{
  "Usuario":"prueba1@factorum.com.mx",
  "Password":"<contraseña>",
  "RFC":"XIA190128J61",
  "XML":"<arreglo byte>"
}
```

Nota: Los datos utilizados son solo con fines de ejemplificar los datos a enviar, si se envían tal cual no se realizará la emisión.

Ejemplo de respuesta Json de petición de timbrado

Timbrado exitosamente


```
{
  "returnStringXML":"<xml string>",
  "returnFileXML":"<arreglo de byte>",
  "returnFileQRCode":"<arreglo de byte>",
  "fechaTimbrado":"2021-04-09T11:05:55",
  "uuid":"B8DE8FBE-7E57-47F2-90C9-2E8F7A2C89F7",
"selloCFD":"m4eRFheFW6NyWXhQMqXqLqGvTlzZYFT5d9hbdHJfq7nUdMZp6hDUc57p+jm+V/0kbS
GMUI+SaFvAVpXqCoxny79CpAjirHNvMejEXytCZKftePlDbuaMi7z2rmi1tN4Lk+2p2z/rvJc/4+EDkiRC6
Gy4tIqsyxiUnjvEQm9WXLMXtZ8OkDaAo5B3IyuSa6Ey+bzQ+QjsjEAxGMKggptEZt+g+TmpdfIc9c6iY/
dkKScijja3WEzj/Y5CQEIJDvww/0awyP32Cf6qMQH6mgiRS1OB1ZtqJxfYFyBQEYLUzMHIbmCWiz5C
7YWvYGFXJU90QWoU8gSHXSDHMyXSSF0pmg==",
  "noCertificadoSAT":"30001000000400002495",
"selloSAT":"FIpAATImtc3WxyZC7fIfjIlsX+aoRNer7+F2Y0TAXqljRVUmb+hq4C/b+SajinVYbHl2pB+S//R
p4Mv28XMwAX+nUk1OryivHQwv7+Hgkz+Oge0eOTHcwDH5fc9fPQt0hB12afQjZRTDIFUFyNSceZ4a
zeESCzuuontZkmv4OrioUC/6UmBWhT0DBEzo9NxLlvubBobve8I4UlipBAnRW2ycD8bo5BCLwze3SR
xqO0pF3VetPNb+EcFkNOlg+9c7BDJyXfCz6KiTRrZf4KpHxIFuxE3kLFlECT6QjL+k27rI0fC5pXOhE7a
4gkPA8N8vMnWjM/LBTonqjUQAZppSLA==",
"cadenaOriginalSello":"||3.3|JS|9327|2021-04-08T13:56:58|01|00001000000408848831|1000.00|MXN
|1160.00|I|PUE|02300|AAA010101AAA|Pruebas SA DE
CV|601|LAN7008173R5|G03|01010101|10|XUN|LOREM|100.00|1000.00|1000.00|002|Tasa|0.160000|
160.00|002|Tasa|0.160000|160.00|160.00||",
"cadenaOriginalTimbre":"||1.1|B8DE8FBE-7E57-47F2-90C9-2E8F7A2C89F7|2021-04-09T11:05:55|E
ME000602QR9|m4eRFheFW6NyWXhQMqXqLqGvTlzZYFT5d9hbdHJfq7nUdMZp6hDUc57p+jm+V/0
kbSGMUI+SaFvAVpXqCoxny79CpAjirHNvMejEXytCZKftePlDbuaMi7z2rmi1tN4Lk+2p2z/rvJc/4+EDkiR
C6Gy4tIqsyxiUnjvEQm9WXLMXtZ8OkDaAo5B3IyuSa6Ey+bzQ+QjsjEAxGMKggptEZt+g+TmpdfIc9c6i
Y/dkKScijja3WEzj/Y5CQEIJDvww/0awyP32Cf6qMQH6mgiRS1OB1ZtqJxfYFyBQEYLUzMHIbmCWiz5
C7YWvYGFXJU90QWoU8gSHXSDHMyXSSF0pmg==|30001000000400002495||",
  "mensajeError":"Documento firmado exitosamente",
  "codigo":200
}
```


Nota: Los datos utilizados son solo con fines de ejemplificar los datos que se recibirán en la respuesta.

Timbrado con error de autenticación


```
{
  "returnStringXML":null,
  "returnFileXML":null,
  "returnFileQRCode":null,
  "fechaTimbrado":"0001-01-01T00:00:00",
  "uuid":null,
  "selloCFD":null,
  "noCertificadoSAT":null,
  "selloSAT":null,
  "cadenaOriginalSello":null,
  "cadenaOriginalTimbre":null,
  "mensajeError":"El usuario no tiene vigente ningún plan de timbrado",
  "codigo":607
}
```


Timbrado Masivo Test

```
String urlApi = "http://api_cloud.factorumweb.com/ApiTimbrado/Timbrado/TimbradoMasivoTest/";
var @params = new Dictionary<string, object>();
@params.Add("Usuario", usuario);
@params.Add("Password", password);
@params.Add("RFC", rfc);
@params.Add("XML", xml);
JsonSerializerSettings settings = new JsonSerializerSettings();
settings.MaxDepth = int.MaxValue;
var inputJson = JsonConvert.SerializeObject(@params, settings);

var content = new StringContent(inputJson, System.Text.Encoding.UTF8, "application/json");

using (HttpClient cliente = new HttpClient())
{
    using (var response = await cliente.PostAsync(urlApi, content))
    {
        if (response.StatusCode != HttpStatusCode.OK)
        {
            return responseFactorum;
        }

        string apiResponse = await response.Content.ReadAsStringAsync();
        responseFactorum = JsonConvert.DeserializeObject<TimbradoMasivoResponse>(apiResponse);
    }
}
return responseFactorum;
```

Obtener Estatus

```
TimbradoMasivoResponse responseFactorum = null;

String urlApi = "http://api_cloud.factorumweb.com/ApiTimbrado/Timbrado/GetStatusTimbradoMasivo/<idpeticion>";

using (HttpClient cliente = new HttpClient())
{
    using (var response = await cliente.GetAsync(urlApi))
    {
        if (response.StatusCode != HttpStatusCode.OK)
        {
            return responseFactorum;
        }

        string apiResponse = await response.Content.ReadAsStringAsync();
        responseFactorum = JsonConvert.DeserializeObject<TimbradoMasivoResponse>(apiResponse);
    }
}
return responseFactorum;
```

En caso de error…

recuerda revisar el mensaje y código que se mencionan en esta documentación.

Si tienes problemas con el servicio o si requieres apoyo en las respuestas recibidas del servicio recuerda que puedes solicitarlo enviando un correo a nuestra Mesa de Soporte: soporte@factorum.com.mx; te pedimos en caso de que lo solicites que proporciones lo siguiente:

1. El método que estás utilizando.
2. El usuario y rfc con el que estás realizando la petición.
3. En caso de que estés utilizando el servicio masivo, la id de solicitud o el zip que estás solicitando emitir.
4. En caso de error también el mensaje y código de respuesta.