



API

Manual de Integración Timbrado CFDI

Tabla de Contenido

Control de versiones	2
Antecedentes.....	3
Objetivos	3
Alcance	3
Servicio API de timbrado CFDI 4.0.....	4
Endpoint Auth.....	4
Token (POST).....	5
Endpoint Emisores	6
Alta Emisor (POST)	6
Emisor (PUT)	7
Cargar Logotipo (POST)	8
Editar Logo (PUT).....	9
Cargar Certificado (POST).....	10
Editar Certificado ARFC (PUT)	11
Cargar Carta Consentimiento (POST)	12
Endpoint Timbrador	13
Endpoint Cancelador.....	16
Firmas de aprobación	¡Error! Marcador no definido.

La información contenida en este documento es confidencial

Este documento que incluye: gráficas, ilustraciones y tablas es confidencial y no debe ser duplicado, usado o difundido en su totalidad o parcialmente para ningún propósito diferente al de evaluación de la propuesta. Este documento puede incluir marcas y nombres comerciales para referirse a las entidades que son propietarias de estos o a sus productos. La empresa renuncia a cualquier interés sobre la propiedad de marcas y nombres comerciales que no sean los suyos.

Control de versiones

Versión	Cambio	Responsable	Fecha de modificación
1.0	Se crea documento	Fernando Morales Chablé	21/07/2022
	Se revisa y documenta	José Jair Tristán López Bosch	25/07/2022
	Se realizan observaciones	Julio Cesar Vera	04/08/2022
1.1	Actualización de Manual	Jonathan Vega Bustamante	23/08/2022
1.2	Actualización de Manual	Jonathan Vega Bustamante	10/03/2023
1.3	Actualización de Manual, se agrega cancelaciones	Gerardo Roque Gutierrez	16/03/2023

Antecedentes

Actualmente nuestro servicio de timbrador 4.0 es mediante un Web Service WCF, el cual nos permite alcanzar un porcentaje de clientes limitados. Tenemos clientes que requieren el uso de tecnologías más ágiles, tales como el consumo de un API REST ful que pueda proveerlos del mismo tipo de servicio.

Objetivos

El objeto de este documento es ofrecer un sistema para los clientes cuyo interés con TurboPac solo sea el proceso de sellado de Timbre Fiscal Digital y validación ante el SAT, sin necesidad de contar con una bóveda de resguardo, proporcionar un servicio de timbrado rápido, efectivo, eficaz con validaciones de estructura y adicionales contemplados por el anexo 20 y 29; al igual un manejo JSON de petición y respuestas para facilitar el manejo de datos.

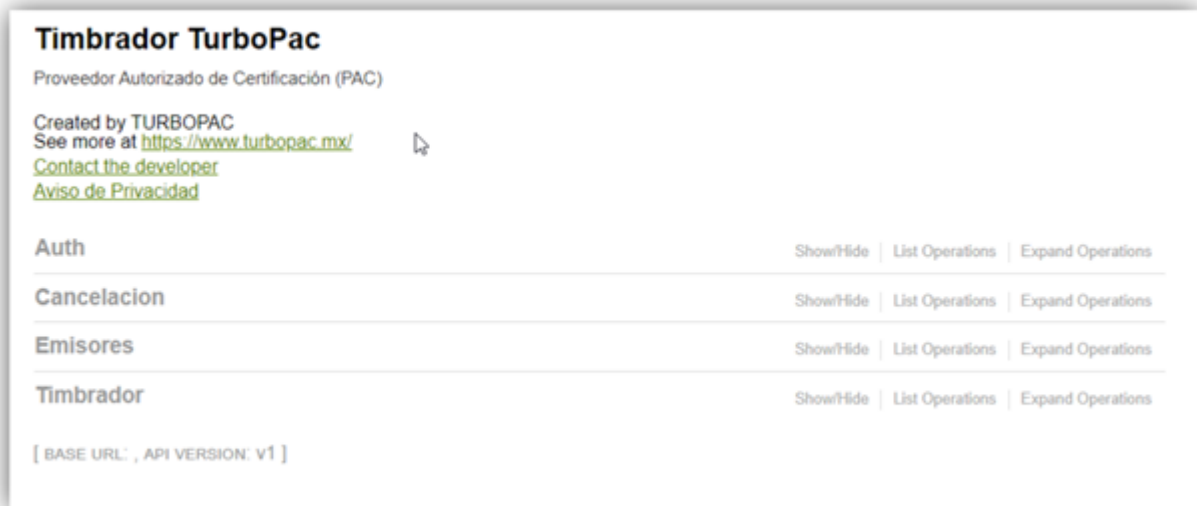
Alcance

Ejecución en tiempo real de la API, recibir una lista de CFDI's generados con estructura de CFDI 4.0, para su sellado y timbrado, así como el alta de emisores y recursos de este.

Servicio API de timbrado CFDI 4.0

Introducción API

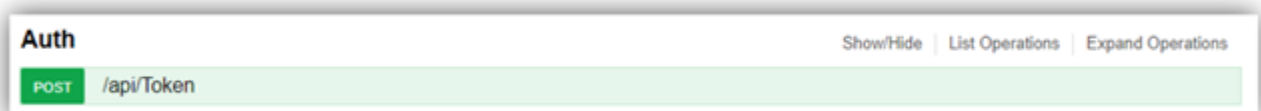
Este servicio proporcionará las siguientes funciones presentadas a continuación que permiten realizar operaciones de emisores, así como el timbrado de un CFDI de la versión 4.0, todas las funciones requieren de un token de acceso que podrán generar en el mismo servicio con las credenciales necesarias.



<https://bovedaemision.turbopacmx.com/TPApiTimbradoV40/swagger/ui/index>

Endpoint Auth

El endpoint de autorización nos permite generar un token de acceso con el que podremos ejecutar cualquiera de las otras funciones del servicio. Las credenciales para obtener el token le corresponden a cada uno de los clientes registrados en la base de datos.



Token (POST)

Genera un token con el cual podremos utilizar las demás funciones expuestas por el servicio. El token expira después de 30 minutos de haber sido generado, información que se visualiza en la respuesta de la petición.

Parámetros

Parameters				
Parameter	Value	Description	Parameter Type	Data Type
username	(required)		query	string
password	(required)		query	string
<button>Try it out!</button>				

Parámetros necesarios para la generación de un token de acceso

Response

```

Response Class (Status 200)
OK
Model Example Value
ResponseToken {
  Access_Token (string, optional),
  Type_Token (string, optional),
  Expires_In (string, optional),
  Cliente (string, optional),
  Start_In (string, optional),
  Errores (Array[ErrorGeneralTurboPAC], optional)
}
ErrorGeneralTurboPAC {
 Codigo (string, optional),
  Mensaje (string, optional)
}
Response Content Type application/json
  
```

Modelo de respuesta correspondiente a la función para generar un token

Endpoint Emisores

Nos permite dar de alta un emisor o editar la información de este, también podemos agregar o editar los certificados con los cuales podremos procesar nuestros CFDI's, agregar o actualizar el logo y cargar la carta consentimiento de este.

Emisores		Show/Hide	List Operations	Expand Operations
POST	/api/AltaEmisor			
PUT	/api/EditarEmisor			
POST	/api/CargarLogotipo			
POST	/api/CargarCertificado			
PUT	/api/EditarLogo			
PUT	/api/EditarCertificadosARFC			
POST	/api/CargarCartaConsentimiento			

Alta Emisor (POST)

Nos permite dar de alta en el sistema un emisor y su información correspondiente como su razón social, régimen fiscal, el domicilio, entre otros datos.

Parámetros

Parameter	Value	Description	Parameter Type	Data Type
access_token	(required)		query	string
request	(required)		body	Model
	Parameter content type: application/json			Example Value <pre>RequestEmisorTPPost { RFCEmisor (string), RazonSocial (string), RegimenFiscal (string), FacAtribAdquiriente (string, optional), Calle (string, optional), NumeroInterior (string, optional), NumeroExterior (string, optional), Colonia (string, optional), DelegacionMunicipio (string, optional), LugarExpedicion (string), Estado (string, optional), Telefono (string, optional), EmailEmisor (string, optional) }</pre>

Try it out!

Parámetros necesarios para dar de alta un nuevo emisor

Response

Response Class (Status 200)
 OK

Model	Example Value
ResponseEmissorTP { RFCEmisor (string, optional), RazonSocial (string, optional), Cliente (string, optional), Valido (boolean, optional), Errores (Array[ErrorGeneralTurboPAC], optional) } ErrorGeneralTurboPAC { Codigo (string, optional), Mensaje (string, optional) }	

Response Content Type:

Modelo de respuesta correspondiente a la función para dar de alta un emisor

Emisor (PUT)

Nos permite editar los datos de un emisor en específico, razón social, régimen fiscal, domicilio, entre otros.

Parámetros

Parameters				
Parameter	Value	Description	Parameter Type	Data Type
access_token	(required)		query	string
request	(required)		body	Model Example Value
	Parameter content type: application/json			RequestEmissorTPPut { RFCEmisor (string), RazonSocial (string, optional), RegimenFiscal (string, optional), FacAttribAdquiriente (string, optional), Calle (string, optional), NumeroInterior (string, optional), NumeroExterior (string, optional), Colonia (string, optional), DelegacionMunicipio (string, optional), LugarExpedicion (string, optional), Estado (string, optional), Telefono (string, optional), EmailEmisor (string, optional) }

Try it out!

Response

Response Class (Status 200)
 OK

Model	Example Value
ResponseEmisorTP { RFCEmisor (string, optional), RazonSocial (string, optional), Cliente (string, optional), Valido (boolean, optional), Errores (Array[ErrorGeneralTurboPAC], optional) } ErrorGeneralTurboPAC { Codigo (string, optional), Mensaje (string, optional) }	

Response Content Type application/json ▼

Modelo de respuesta correspondiente a la función para editar un emisor

Cargar Logotipo (POST)

Nos permite agregar un logotipo (el logotipo debe estar en base 64) a un emisor en específico, para poder visualizarlo en su portal y sus representaciones impresas (esta funcionalidad es opcional).

Parámetros

Parameters				
Parameter	Value	Description	Parameter Type	Data Type
access_token	(required)		query	string
rfcEmisor	(required)		query	string
imagenBase64	(required)		query	string
Try it out!				

Parámetros necesarios para registrar el logo de un emisor en específico

Response

Response Class (Status 200)
OK

Model	Example Value
ResponseEmissorTP { RFCEmisor (string, optional), RazonSocial (string, optional), Cliente (string, optional), Valido (boolean, optional), Errores (Array[ErrorGeneralTurboPAC], optional) }	
ErrorGeneralTurboPAC { Codigo (string, optional), Mensaje (string, optional) }	

Response Content Type

Modelo de respuesta correspondiente a la función para registrar un logotipo de un emisor en específico

Editar Logo (PUT)

Nos permite sustituir el logotipo (el logotipo debe estar en base 64) de un emisor en específico, para poder visualizarlo en su portal y sus representaciones impresas (esta funcionalidad es opcional).

Parámetros

Parameter	Value	Description	Parameter Type	Data Type
access_token	(required)		query	string
rfcEmisor	(required)		query	string
imagenBase64	(required)		query	string

Parámetros necesarios para sustituir el logotipo de un emisor en específico

Response

Response Class (Status 200)

OK

Model	Example Value
ResponseEmissorTP { RFCEmissor (string, optional), RazonSocial (string, optional), Cliente (string, optional), Valido (boolean, optional), Errores (Array[ErrorGeneralTurboPAC], optional) } ErrorGeneralTurboPAC { Codigo (string, optional), Mensaje (string, optional) }	

Response Content Type

Modelo de respuesta correspondiente a la función para sustituir el logotipo de un emisor en específico

Cargar Certificado (POST)

Nos permite cargar el certificado de un emisor en específico para poder procesar CFDI's de forma correcta, el certificado es validado antes de ser cargado, el certificado y la llave deben de estar en base 64.

Parámetros

Parameters				
Parameter	Value	Description	Parameter Type	Data Type
access_token	(required)		query	string
rfcEmissor	(required)		query	string
certificadoBase64	(required)		query	string
keyBase64	(required)		query	string
passwordCertificado	(required)		query	string
Try it out!				

Parámetros necesarios para la carga de un certificado

Response

Response Class (Status 200)
OK

Model	Example Value
ResponseEmissorTP { RFCEmissor (string, optional), RazonSocial (string, optional), Cliente (string, optional), Valido (boolean, optional), Errores (Array[ErrorGeneralTurboPAC], optional) } ErrorGeneralTurboPAC { Codigo (string, optional), Mensaje (string, optional) }	

Response Content Type

Modelo de respuesta correspondiente a la función para la carga de un certificado

Editar Certificado ARFC (PUT)

Nos permite editar el certificado cargado de un emisor en específico para poder procesar CFDI's de forma correcta, el certificado es validado antes de ser sustituido, el certificado y la llave deben de estar en base 64.

Parámetros

Parameters				
Parameter	Value	Description	Parameter Type	Data Type
access_token	(required)		query	string
rfcEmissor	(required)		query	string
certificadoBase64	(required)		query	string
keyBase64	(required)		query	string
passwordCertificado	(required)		query	string
Try it out!				

Parámetros necesarios para la sustitución de un certificado

Response

Response Class (Status 200)

OK

Model	Example Value
ResponseEmisorTP { RFCEmisor (string, optional), RazonSocial (string, optional), Cliente (string, optional), Valido (boolean, optional), Errores (Array[ErrorGeneralTurboPAC], optional) } ErrorGeneralTurboPAC { Codigo (string, optional), Mensaje (string, optional) }	

Response Content Type

Modelo de respuesta correspondiente a la función para la sustitución de un certificado

Cargar Carta Consentimiento (POST)

Nos permite generar la carta consentimiento de un emisor en específico, para así poder realizar operaciones validas con los certificados dados de alta para el emisor, el certificado y la llave deben estar en base 64.

Parámetros

Parameters				
Parameter	Value	Description	Parameter Type	Data Type
access_token	(required)		query	string
rfcEmisor	(required)		query	string
representante	(required)		query	string
certificadoFIEL64	(required)		query	string
llaveFIEL64	(required)		query	string
passwordFIEL	(required)		query	string
Try it out!				

Parámetros necesarios para generar la carta consentimiento

Response

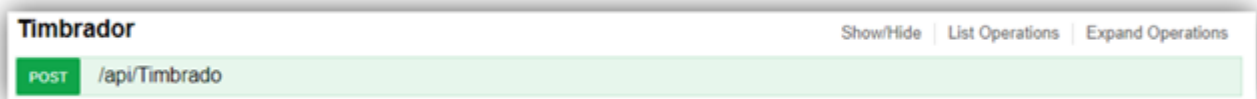
```

Response Class (Status 200)
OK
Model | Example Value
ResponseEmisorTP {
  RFCEmisor (string, optional),
  RazonSocial (string, optional),
  Cliente (string, optional),
  Valido (boolean, optional),
  Errores (Array[ErrorGeneralTurboPAC], optional)
}
ErrorGeneralTurboPAC {
  Codigo (string, optional),
  Mensaje (string, optional)
}
Response Content Type application/json
  
```

Modelo de respuesta correspondiente a la función para generar la carta consentimiento

Endpoint Timbrador

El endpoint de timbrador nos permite timbrar de manera masiva CFDI's de la versión 4.0.



Timbrado (POST)

Nos permite sellar y timbrar de manera masiva CFDI's de la versión 4.0, los CFDI's proporcionados deben de estar en base 64.

Parámetros

Parameters

Parameter	Value	Description	Parameter Type	Data Type
access_token	(required)		query	string
Request	(required) Parameter content type: application/json		body	Model Example Value <pre> RequestSingle { IdPeticion (string, optional), Fecha (string, optional), Contador (integer, optional), Facturas (Array[JsonFacturaSingle], optional) } JsonFacturaSingle { base64 (string, optional), idFile (integer, optional), contador (integer, optional) } </pre>

Try it out!

Parámetros necesarios para la emisión de CFDI's

Response

Response Class (Status 200)

OK

Model	Example Value
-------	---------------

ResponseSingle {

Referencia (string, optional),
 Fecha (string, optional),
 Contador (integer, optional),
 Facturas (Array[FacturaSingle], optional),
 CifrasControl (CifrasControl, optional)

}

FacturaSingle {

Contador (integer, optional),
 XMLTimbrado (string, optional),
 ArchivoQR (string, optional),
 UUID (string, optional),
 FechaSello (string, optional),
 SelloDigital (string, optional),
 NumeroCertificado (string, optional),
 ReporteIncidentes (ReporteIncidentes, optional)

}

CifrasControl {

TotalRecibidos (integer, optional),
 TotalCorrectos (integer, optional),
 TotalIncidentes (integer, optional)

}

ReporteIncidentes {

RfcEmisor (object, optional),
 RfcReceptor (object, optional),
 Incidentes (Array[string], optional)

}

Modelo de respuesta correspondiente a la función la emisión de CFDI's

Endpoint Cancelador

Cancelar (Post)

En este endpoint se podrá realizar la cancelación de cualquier factura, sin la necesidad de que se timbre con el endpoint anterior, por lo cual, arrojará la respuesta sin ningún problema.

POST /api/Cancelar

Parámetros

Al igual que los anteriores endpoint, requerimos de un token para poder realizar dicha acción.

Cancelacion
Show/Hide | List Operations | Expand Operations

POST /api/Cancelar

Response Class (Status 200)
 OK

Model | Example Value

```

{
  "Code": "string",
  "Message": "string",
  "Recib": "string",
  "Success": true,
  "ErrorMessage": "string"
}

```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
access_token	(required)		query	string
request	<pre> { "RFCEmisor": "string", "Motivo": "string", "UUID": "string", "UUIDSustitucion": "string", "ErrorRelacionado": "string", "XML": "string", "PFX": "string", "PassPFX": "string" } </pre>		body	Model Example Value <pre> { "RFCEmisor": "string", "Motivo": "string", "UUID": "string", "UUIDSustitucion": "string", "ErrorRelacionado": "string", "XML": "string", "PFX": "string", "PassPFX": "string" } </pre>

Parameter content type:
application/json

Try it out!

Response

Response Class (Status 200)	
OK	
Model	Example Value
	<pre>{ "Code": "string", "Message": "string", "Recib": "string", "Suces": true, "ErrorMessage": "string" }</pre>

Ejemplo

Request URL
https://localhost:44356/api/Cancelar
Response Body
<pre>{ "Code": null, "Message": "El Cliente no esta activoEl RCemisor no cuenta con un Email registradoEste rfc no esta ligado a un Emisor registrado", "Recib": null, "Suces": false, "ErrorMessage": null }</pre>
Response Code
200

Contáctanos

Ventas



WhatsApp
55 3251 8542



Teléfono
55 4161 0838



Email
contacto@qrea-t.mx



Soporte Operativo



WhatsApp
55 7498 9419



Teléfono
55 4161 0885



Email
mesadeayuda@turbopac.biz



¡Estamos para apoyarte!