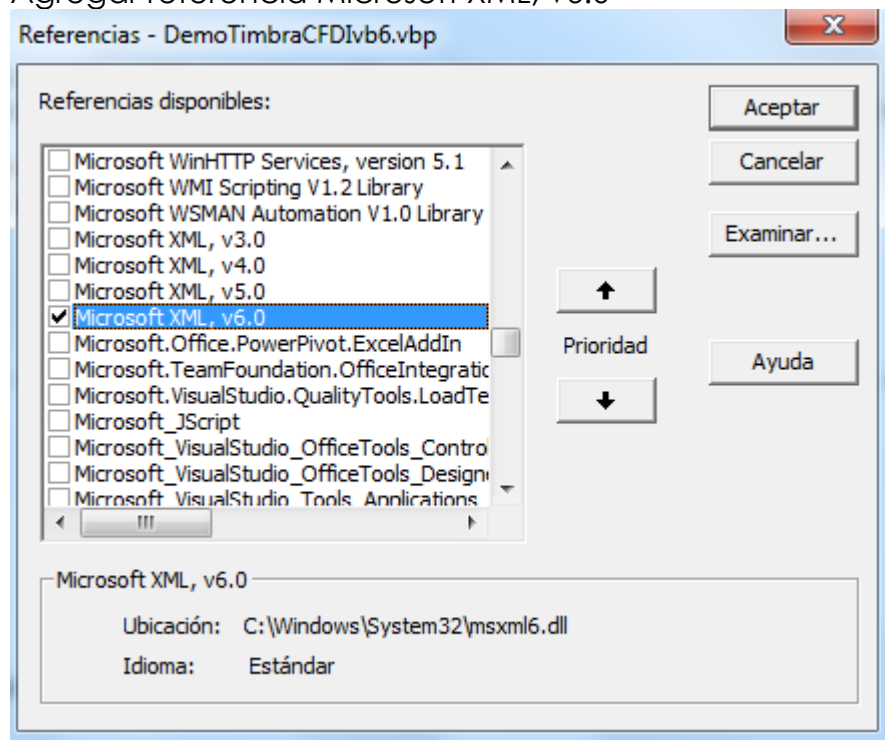


Proyecto: Integración de Timbrado usando Visual Basic 6.0

Documento: Manual de implementación

Referencias necesarias

Agregar referencia Microsoft XML, v6.0



Definiciones de métodos

Es necesario agregar las definiciones de los métodos que vamos a consumir, ya que se van a enviar con los parámetros al servicio web, estas definiciones las tomamos de lo mostrado en el explorador, al seleccionar cada uno de los métodos.

Timbrado

Click [here](#) for a complete list of operations.

TimbraCFDI

Servicio de timbrado para Integradores

Test

The test form is only available for requests from the local machine.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /servicioIntegracionpruebas/timbrado.asmx HTTP/1.1
Host: 201.178.12.89
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://localhost/TimbraCFDI"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <TimbraCFDI xmlns="http://localhost/">
      <usuarioIntegrador>string</usuarioIntegrador>
      <xmlComprobanteBase64>string</xmlComprobanteBase64>
      <idComprobante>string</idComprobante>
    </TimbraCFDI>
  </soap:Body>
</soap:Envelope>
```

METODO Registra Emisor

```
Private Const cSOAPREGEMISOR = "<?xml version=""1.0"" encoding=""utf-8""?>" & _
"<soap:Envelope xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance"" _
xmlns:xsd=""http://www.w3.org/2001/XMLSchema"" _
xmlns:soap=""http://schemas.xmlsoap.org/soap/envelope/"">" & _
"<soap:Body>" & _
"<RegistraEmisor xmlns=""http://localhost/"">" & _
"<usuarioIntegrador>string</usuarioIntegrador>" & _
"<rfcEmisor>string</rfcEmisor>" & _
"<base64Cer>string</base64Cer>" & _
"<base64Key>string</base64Key>" & _
"<contrasena>string</contrasena>" & _
"</RegistraEmisor>" & _
"</soap:Body>" & _
"</soap:Envelope>"
```

METODO TimbraCFDI

```
Private Const cSOAPTIMBRACFDI = "<?xml version='1.0' encoding='utf-8'?" & _  
"<soap:Envelope xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'"  
xmlns:xsd='http://www.w3.org/2001/XMLSchema'"  
xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'">" & _  
"<soap:Body>" & _  
"<TimbraCFDI xmlns='http://localhost/'">" & _  
"<usuarioIntegrador>string</usuarioIntegrador>" & _  
"<xmlComprobanteBase64>string</xmlComprobanteBase64>" & _  
"<idComprobante>string</idComprobante>" & _  
"</TimbraCFDI>" & _  
"</soap:Body>" & _  
"</soap:Envelope>"
```

METODO CancelaCFDI

```
Private Const cSOAPCANCELACFDI = "<?xml version='1.0' encoding='utf-8'?" & _  
"<soap:Envelope xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'"  
xmlns:xsd='http://www.w3.org/2001/XMLSchema'"  
xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'">" & _  
"<soap:Body>" & _  
"<CancelaCFDI xmlns='http://localhost/'">" & _  
"<usuarioIntegrador>string</usuarioIntegrador>" & _  
"<rfcEmisor>string</rfcEmisor>" & _  
"<folioUUID>string</folioUUID>" & _  
"</CancelaCFDI>" & _  
"</soap:Body>" & _  
"</soap:Envelope>"
```

METODO ObtieneCFDI

```
Private Const cSOAPOBTIENECFDI = "<?xml version='1.0' encoding='utf-8'?" & _  
"<soap:Envelope xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'"  
xmlns:xsd='http://www.w3.org/2001/XMLSchema'"  
xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'">" & _  
"<soap:Body>" & _  
"<ObtieneCFDI xmlns='http://localhost/'">" & _  
"<usuarioIntegrador>string</usuarioIntegrador>" & _  
"<rfcEmisor>string</rfcEmisor>" & _  
"<folioUUID>string</folioUUID>" & _  
"</ObtieneCFDI>" & _  
"</soap:Body>" & _  
"</soap:Envelope>"
```

1. TimbraCFDI

```
Dim parser As New DOMDocument
Dim parserRespuesta As New DOMDocument
Dim xmlComprobanteBase64 As String
Dim idcomprobante As Long
'limpia campos
limpia
txtffiscalo.Text = ""
txtffiscalc.Text = ""

'metodo del servicio web
strSOAPAction = "http://localhost/TimbraCFDI"

'archivo sin timbrar de prueba
xmlComprobanteBase64 = ReadBinData(App.Path & "\archivos\comprobanteSinTimbrar.xml")

'Para efectos de prueba utilizaremos un numero aleatorio
idcomprobante = (100000) * CInt(1000 * Rnd)

' cargar el código SOAP para TIMBRACFDI
parser.loadXML cSOAPTIMBRACFDI

' Indicar los parámetros a enviar
parser.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDI/usuarioIntegrador").Text = usuarioIntegrador
parser.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDI/xmlComprobanteBase64").Text =
xmlComprobanteBase64
parser.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDI/idComprobante").Text = idcomprobante

'Envía la petición de WebService
If InvokeWebService(parser.xml, strSOAPAction, strWsd1, parserRespuesta) Then
    Dim tipoExcepcion, numError, descResultado, xml, cod64 As String
    'tipo de excepción
    tipoExcepcion =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult").childNodes.
Item(0).Text
    'Numero de error
    numError =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult").childNodes.
Item(1).Text
    'Descripcion del resultado
    descResultado =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult").childNodes.
Item(2).Text
    'xml timbrado
    xml =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult").childNodes.
Item(3).Text
    'codigo base64 QR
    cod64 =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult").childNodes.
Item(4).Text
    'Cadena original SAT
    txtcodOr.Text =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult").childNodes.
Item(5).Text
```

```

'Validamos si existe error
If numError = 0 Then
    'Obtenemos los datos del xml regresado timbrado
    Dim listaNodos As MSXML2.IXMLDOMNodeList
    Set listaNodos =
parserRespuesta.selectNodes("soap:Envelope/soap:Body/TimbraCFDIResponse/TimbraCFDIResult/anyType")

    parserRespuesta.loadXML Replace(listaNodos.Item(3).Text, "<", "<", , , vbTextCompare)
    parserRespuesta.loadXML Replace(listaNodos.Item(3).Text, ">", ">", , , vbTextCompare)
    'Mostramos los datos obtenidos
    txtsdigital.Text =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:TimbreFiscalDigital/@selloSAT").Text
    txtcersat.Text =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:TimbreFiscalDigital/@noCertificadoSAT").Text
    txttimsan.Text =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:TimbreFiscalDigital/@FechaTimbrado").Text
    txtfolsat.Text =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:TimbreFiscalDigital/@UUID").Text
    'Guarda XML timbrado
    parserRespuesta.save App.Path & "\Facturas\" & idcomprobante & "Timbre.xml"
    'Función genera QR de código base64
    Call generaQR(cod64, idcomprobante & "QR.jpg", qr)
    listaNodos.Reset
    resultado.Text = "El comprobante fue timbrado exitosamente"
Else
    resultado.Text = descResultado
End If
Set parserRespuesta = Nothing
End If
Set parser = Nothing

```

2. ObtieneCFDI

```

Dim parser As New XmlDocument
Dim parserRespuesta As New XmlDocument
'limpia campos
limpia
Form1.txtffiscalc.Text = ""

'Metodo del servicio web
strSOAPAction = "http://localhost/ObtieneCFDI"

' cargar el código SOAP para OBTIENECDI
parser.loadXML cSOAPOBTIENECDI

'Folio Fiscal
folioUUID = txtffiscalc.Text

' Indicar los parámetros a enviar
parser.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDI/usuarioIntegrador").Text = usuarioIntegrador
parser.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDI/rfcEmisor").Text = rfcEmisor
parser.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDI/folioUUID").Text = folioUUID

'Envía la petición de WebService
If InvokeWebService(parser.xml, strSOAPAction, strWSDL, parserRespuesta) Then
    Dim tipoExcepcion, numError, descResultado, xml, cod64 As String
    'tipo de excepción
    tipoExcepcion =
    parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDIResponse/ObtieneCFDIResult").childNodes(0).Text
    'Numero de error
    numError =
    parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDIResponse/ObtieneCFDIResult").childNodes(1).Text
    'Descripcion del resultado
    descResultado =
    parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDIResponse/ObtieneCFDIResult").childNodes(2).Text
    'XML
    xml =
    parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDIResponse/ObtieneCFDIResult").childNodes(3).Text

    'codigo base64 QR
    cod64 =
    parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDIResponse/ObtieneCFDIResult").childNodes(4).Text
    'Cadena original SAT
    txtcodOrO.Text =
    parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/ObtieneCFDIResponse/ObtieneCFDIResult").childNodes(5).Text
    'Validamos si existe error
    If numError = 0 Then
        'Obtenemos los datos del xml regresado timbrado
        Dim listaNodos As MSXML2.IXMLDOMNodeList
        Set listaNodos =
        parserRespuesta.selectNodes("/soap:Envelope/soap:Body/ObtieneCFDIResponse/ObtieneCFDIResult/anyType")
    
```

```

    parserRespuesta.loadXML Replace(listaNodos.Item(3).Text, "<", "<", , , vbTextCompare)
    parserRespuesta.loadXML Replace(listaNodos.Item(3).Text, ">", ">", , , vbTextCompare)
    'Mostramos los datos obtenidos
    txtsdigital0.Text =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:TimbreFiscalDigital/@selloSAT").Text
    txtcersat0.Text =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:TimbreFiscalDigital/@noCertificadoSAT").Text
    txttimsan0.Text =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:TimbreFiscalDigital/@FechaTimbrado").Text
    txtfolsat0.Text =
parserRespuesta.selectSingleNode("cfdi:Comprobante/cfdi:Complemento/tfd:TimbreFiscalDigital/@UUID").Text
    'Guarda xml obtenido
    parserRespuesta.save App.Path & "\CFDI\" & folioUUID & "_CFDI.xml"

    'Función genera QR de código base64
    qr0.Scale

    Call generaQR(cod64, folioUUID & "QR.jpg", qr0)
    listaNodos.Reset
    resultado.Text = "Se recupero correctamente CFDI"
Else
    resultado.Text = descResultado
End If
End If
Set parser = Nothing

```

3. CancelaCFDI

```

Dim parser As New DOMDocument
Dim parserRespuesta As New DOMDocument

'Metodo
strSOAPAction = "http://localhost/CancelaCFDI"

'cargar el código SOAP para CANCELACFDI
parser.LoadXml cSOAPCANCELACFDI

'Indicar los parámetros a enviar
parser.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDI/usuarioIntegrador").Text = usuarioIntegrador
parser.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDI/rfcEmisor").Text = rfcEmisor
parser.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDI/folioUUID").Text = txtffiscalc.Text

'Envía la petición de WebService
If InvokeWebService(parser.xml, strSOAPAction, strWsdL, parserRespuesta) Then
    Dim tipoExcepcion, numError, descResultado, xml As String
    'tipo de excepción
    tipoExcepcion =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDIResponse/CancelaCFDIResult").childNodes.Item(0).Text
    'Numero de error
    numError =
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDIResponse/CancelaCFDIResult").childNodes.Item(1).Text
    'Descripcion del resultado
    descResultado =

```

```
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDIResponse/CancelaCFDIResult").childNodes.  
s.Item(2).Text  
    'XML  
    xml =  
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/CancelaCFDIResponse/CancelaCFDIResult").childNodes.  
s.Item(3).Text  
    resultado.Text = descResultado  
End If
```

4. RegistraEmisor

```
Dim parser As New DOMDocument  
Dim parserRespuesta As New DOMDocument  
Dim archivoCerBase64 As String  
Dim archivoKeyBase64 As String  
'Metodo  
strSOAPAction = "http://localhost/RegistraEmisor"  
  
'Convierte los archivos a base64  
archivoCerBase64 = ReadBinData(App.Path & "\archivos\aaa010101aaa__csd_01.cer")  
archivoKeyBase64 = ReadBinData(App.Path & "\archivos\aaa010101aaa__csd_01.key")  
'cargar el código SOAP para REGISTRA EMISOR  
parser.LoadXml cSOAPREGEMISOR  
  
'Indicar los parámetros a enviar  
parser.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisor/usuarioIntegrador").Text =  
usuarioIntegrador  
parser.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisor/rfcEmisor").Text = rfcEmisor  
parser.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisor/base64Cer").Text = archivoCerBase64  
parser.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisor/base64Key").Text = archivoKeyBase64  
parser.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisor/contrasena").Text = contrasena  
  
'Envía la petición de WebService  
If InvokeWebService(parser.xml, strSOAPAction, strWsd1, parserRespuesta) Then  
    Dim tipoExcepcion, numError, descResultado As String  
    'tipo de excepción  
    tipoExcepcion =  
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisorResponse/RegistraEmisorResult").chi  
ldNodes.Item(0).Text  
    'Numero de error  
    numError =  
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisorResponse/RegistraEmisorResult").chi  
ldNodes.Item(1).Text  
    'Descripcion del resultado  
    descResultado =  
parserRespuesta.selectSingleNode("/soap:Envelope/soap:Body/RegistraEmisorResponse/RegistraEmisorResult").chi  
ldNodes.Item(2).Text  
    resultado.Text = descResultado  
End If
```